



PF-3200/118/US/PTO

UNI.038

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re patent application of

Tsugio Okamoto

Serial No.: 10/623,644

Group Art Unit: Not Yet Assigned

Filing Date: July 22, 2003

Examiner: Unknown

For: ADDRESS RETRIEVAL APPARATUS

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

SUBMISSION OF PRIORITY DOCUMENT

Sir:

Submitted herewith is a certified copy of Japanese Application Number 2002-214020 filed on July 23, 2002, upon which application the claim for priority is based.

Respectfully submitted,

Sean M. McGinn
Registration No. 34,386

Date: 10/22/03

McGinn & Gibb, PLLC
Intellectual Property Law
8321 Old Courthouse Road, Suite 200
Vienna, VA 22182-3817
(703) 761-4100
Customer No. 21254

日 本 国 特 許 庁
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office

出 願 年 月 日

Date of Application:

2002年 7月23日

出 願 番 号

Application Number:

特願2002-214020

[ST.10/C]:

[JP2002-214020]

出 願 人

Applicant(s):

日本電気株式会社

2003年 6月 2日

特 許 庁 長 官
Commissioner,
Japan Patent Office

太田信一郎

出証番号 出証特2003-3041789

【書類名】 特許願

【整理番号】 40410692PY

【あて先】 特許庁長官殿

【国際特許分類】 H04L 12/00
H04L 12/56

【発明者】

【住所又は居所】 東京都港区芝五丁目 7 番 1 号 日本電気株式会社内

【氏名】 岡本 継男

【特許出願人】

【識別番号】 000004237

【氏名又は名称】 日本電気株式会社

【代理人】

【識別番号】 100083987

【弁理士】

【氏名又は名称】 山内 梅雄

【手数料の表示】

【予納台帳番号】 016252

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 9006535

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 アドレス検索装置

【特許請求の範囲】

【請求項 1】 通信ネットワークから次のルートに振り分けるべきデータ信号を入力し、そのデータ信号の最終的な宛先に至るルートを表わしたビット列からなるルート情報を予め定めたビット列ずつ先頭から区切ってそれぞれの階層の分割ビット列として順に取り出す分割ビット列取り出し手段と、

各ルート情報に対応した前記データ信号の振り分け先を示す出力先振り分け情報を記憶した出力先振り分け情報記憶手段と、

前記分割ビット列取り出し手段によって取り出す可能性のある分割ビット列をそれぞれ対応する階層に配置したときに形成される木構造の各節に相当する位置にそれぞれ検索テーブルノードを配置し、これらの検索テーブルノードの各々に、対応する分割ビット列から次の検索テーブルノードを指示する次検索テーブルノード指示情報あるいは次の検索テーブルノードが存在しないときには前記出力先振り分け情報記憶手段における前記データ信号の振り分け先を示す出力先振り分け情報が記載されている前記出力先振り分け情報記憶手段の該当箇所を指示する出力先振り分け情報格納箇所指示情報を記憶した検索テーブルと、

前記分割ビット列取り出し手段によって取り出した分割ビット列を用いて、第 1 の階層の検索テーブルノードから次検索テーブルノード指示情報が読み出される間、分割ビット列を順に取り出しながら検索テーブルノードを 1 階層ずつ階層を高めて検索を行い、出力先振り分け情報格納箇所指示情報が検索された時点でこの階層の移動を停止して検索する階層を確定する階層確定手段と、

この階層確定手段によって確定した検索テーブルノードから読み出された出力先振り分け情報格納箇所指示情報を用いて前記出力先振り分け情報記憶手段から前記データ信号についての出力先振り分け情報を読み出す出力先振り分け情報読み出し手段

とを具備することを特徴とするアドレス検索装置。

【請求項 2】 前記検索テーブルノードのうちで次検索テーブルノード指示情報を格納しているもののみが、前記次検索テーブルノード指示情報を格納する

領域を配置していることを特徴とする請求項 1 記載のアドレス検索装置。

【請求項 3】 各階層の検索テーブルノードは、そのテーブルノードに連結する次の階層の検索テーブルノードが存在するか否かを示す有効情報有無情報を配置していることを特徴とする請求項 1 記載のアドレス検索装置。

【請求項 4】 前記出力先振り分け情報格納箇所指示情報は、出力先振り分け情報記憶手段における出力先振り分け情報が記憶されている領域と、その領域における出力先振り分け情報の格納箇所を示す番地とで構成されていることを特徴とする請求項 1 記載のアドレス検索装置。

【請求項 5】 前記出力先振り分け情報記憶手段における出力先振り分け情報が記憶されている領域は、対応する分割ビット列を構成するビット数の二乗の数の番地を備えていることを特徴とする請求項 4 記載のアドレス検索装置。

【請求項 6】 前記分割ビット列取り出し手段が順に取り出す分割ビット列は、それぞれ互いに等しいビット数で構成されていることを特徴とする請求項 1 記載のアドレス検索装置。

【発明の詳細な説明】

【0 0 0 1】

【発明の属する技術分野】

本発明は通信ネットワークでパケット信号等のデータ信号の送出先としてのアドレスを検索するアドレス検索装置に係わり、たとえばインターネットでパケット信号の振り分けを行うルータにおける振り分け先の検索処理に好適なアドレス検索装置に関する。

【0 0 0 2】

【従来の技術】

インターネットに代表される通信ネットワークを使用した各種通信が広範囲に行われている。このような通信ネットワークで複数のネットワークを接続するための装置の 1 つとしてルータがある。ルータは、IP (Internet Protocol) パケット信号等のデータ信号の送出先を示すルート情報を検索することで、その受信したデータ信号をどの出力ポートから出力させるかを決定するようになっている。そして、ネットワーク上に複数配置されたそれぞれのルータがこのように受

信したデータ信号の出力先を次々と決定し出力先を振り分けていくことで経路が設定されていき、そのデータ信号がその経路に沿って最終的な宛先に転送されることになる。

【 0 0 0 3 】

このルーティングの様子は、ちょうど電話番号の最初の数字から順にそれぞれ対応する交換機が出力先を振り分けていき、最終的にその電話番号の全桁と一致する相手先が発呼されるのと似ている。

【 0 0 0 4 】

このため、インターネット等の通信ネットワークでデータ信号の振り分けを行う各ルータは、入力されたデータ信号のルート情報を基にして、そのデータ信号がそのルータの複数の出力ポートの中のいずれから出力されるべきかを決定し、決定した出力ポートからデータ信号を送出することになる。このような出力ポートの選択動作を可能にするために、個々のルータは検索テーブルを備えており、この検索結果に応じて出力ポートの選択を行っている。

【 0 0 0 5 】

ところで通信ネットワークでは、データ信号を受信する端末が追加されたり削除されるといった変化が生じる場合がある。たとえば今まで A という送出先を示すルート情報がきたときにそのルータが第 1 の出力ポートを選択してデータ信号を転送していたのに、第 1 の出力ポート側のある端末が削除されて、そのルートを辿っても送出先に行き着かないような場合である。このような場合には、たとえば第 3 の出力ポートを経て、迂回するような形で相手先にデータ信号を送信するといったようなルートの変更が行われるような場合がある。このような場合に対処するために、ネットワークの構成に変更が生じたときには、個々のルータに備えられている検索テーブルに新たなデータを登録したり、既存のデータの削除あるいは変更が行われる。

【 0 0 0 6 】

データ信号が I P パケット信号である場合を例にとって説明することにする。ルータに備えられる検索テーブルには、I P パケット信号の次の転送先としての「出力インタフェースまたは次の中継先であるルータの I P アドレス」といった

出力ポートを選択するための出力先振り分け情報が、データ信号の有する「IPサブネットアドレス」および「プレフィックス長」として登録されている。

【0007】

ここでサブネットとは、ネットワークを複数のネットワークに分割したものをいう。たとえばIPアドレスはネットワークアドレスとホスト番号から構成されており、ホスト番号の上位の所定ビットがサブネット番号としてネットワークを分割するために用いられている。プレフィックス (Prefix) 長とは、アドレスのうち有効ビット列の長さをビット数で表わしたもの (有効ビット列長) である。

【0008】

検索テーブルの検索を行うにあたっては、検索テーブルに登録されたものの中から、最終的にロングストプレフィックスマッチ (Longest Prefix Match) 法により、IPアドレスと最も長く一致する「IPサブネットアドレス」を決定するようにしている。すなわち、ロングストプレフィックスマッチ法では、登録されているIPアドレスのうち、検索キーのIPアドレスと一致する有効ビット列が最も長いものを選択する。たとえばビット長が3ビットのIPアドレスとして、ビット列“0**”および“00*”の2つが登録されているものとする。ここでビット列“0**”は、先頭ビットの“0”のみが有効ビット値で、2番目以降の“**”は、“0”または“1”の任意の値を表わしている。また、ビット列“00*”は、1ビット目および2ビット目の“00”が有効ビット値で、3番目の“*”は、“0”または“1”の任意の値を表わしている。

【0009】

検索テーブルによる検索では、転送されてきたIPパケットの最終的な送信先を示す、検索キーとしてのIPアドレスと、登録されているこれら2つの登録IPアドレスとをそれぞれ照合する。検索キーのIPアドレスを“0001…”とした場合、ビット列“00*”の方が“0**”よりも、登録IPアドレスと一致するプレフィックス長が長い。したがって、この場合、次の転送先のIPアドレスとしてビット列“00*”が選択されることになる。

【0010】

ところで、IPアドレスのビット数が、この例のように3ビット程度と短い場

合には、その検索も容易である。しかしながら実際に使用される I P アドレスのビット数は、これよりも遥かに長い。例えば、I P プロトコルとしての I P v 4 (Internet Protocol version 4) におけるアドレス長は、3 2 ビットとなっている。近年標準化された他の I P プロトコルとしての I P v 6 (Internet Protocol version 6) では、アドレス長が 1 2 8 ビットと一層長くなっている。このため、ルータで I P アドレスを基にして出力先振り分け情報を得るための検索処理の効率化が問題となる。

【 0 0 1 1 】

従来のこのような検索手法としては、バイナリ・ツリー (Binary Tree) 検索方法とプレフィックス拡張 (Prefix Expansion) 検索方法が存在している。バイナリ・ツリー検索方法は、2 分木データ構造を利用した手法である。

【 0 0 1 2 】

図 1 7 および図 1 8 は、バイナリ・ツリー検索法の概念を説明するためのものである。このうち図 1 7 は 8 つの I P アドレス $P_1 \sim P_8$ のビットパターンを表わしている。先に説明したように符号 “*” は、“0” または “1” の任意の値を表わしているものとする。

【 0 0 1 3 】

図 1 8 はバイナリ・ツリー検索法を示している。この例では、登録 I P アドレス $P_1 \sim P_8$ がルートの起点 1 0 1 を基にツリー構造を構成している。バイナリ・ツリー検索方法では、先頭から 1 ビット単位でビット列のビット値の比較を順番に行う。このため、ツリーを構成する各ノードは、この図に示すようにビット列の各ビット値 (“0” または “1”) にそれぞれ対応している。

【 0 0 1 4 】

特開平 1 1 - 1 9 1 7 8 1 号公報には、ハードウェア回路でバイナリ・ツリーの検索を実現するようにしたアドレス検索装置が開示されている。この提案のバイナリ・ツリー検索方法では、検索を高速化するために必要のない枝を除去した木構造で経路テーブルを構成するようにしている。更にこの検索テーブルをハードウェアを用いて検索することで高速な経路検索を可能にしている。

【 0 0 1 5 】

しかしながら特開平 1 1 - 1 9 1 7 8 1 号公報に開示されたバイナリ・ツリー検索方法では、木構造の形態によって最大アドレスのビット数の分岐が発生することがあり得る。たとえば I P v 4 の場合には、図 1 8 に示した 8 ビット構成の I P アドレスとは異なり、3 2 ビットの I P アドレスを採用している。このため、図 1 8 よりも複雑な木構造となって、最大で 3 2 回の分岐が発生することになる。このように最大で 3 2 回の分岐が発生すると、最大でこの分岐の数だけノードへのアクセスが必要になる。このため、バイナリ・ツリー検索方法はノードへのアクセス数が多くなる可能性があって、検索の効率が悪いという欠点を持つ。特開平 1 1 - 1 9 1 7 8 1 号公報では、アドレス検索装置あるいはアドレス検索方法をソフトウェアで実現するのではなく高速処理に適したハードウェアで実現している。しかしながら、たとえハードウェアで回路構成を行ったとしても、ノードへのアクセスが多ければ検索の効率はソフトウェアと手法の違いによる差ということになって、本質的な改善にはならない。そこで、分岐の回数自体を減少させるためにプレフィックス拡張検索方法を使用したアドレス検索装置あるいはアドレス検索方法が提案されている。

【 0 0 1 6 】

図 1 9 は、プレフィックス拡張検索方法を説明するためのものである。特開 2 0 0 0 - 3 0 7 6 4 1 号公報等で提案されているプレフィックス拡張検索方法では、I P アドレスのビット列を複数の分割ビット列に分割する。そして、分割後のこれら分割ビット列を階層化して、階層化された分割ビット列ごとに検索テーブル 1 1 1 ~ 1 1 4 を設けるようにしている。そして、各検索テーブル 1 1 1 ~ 1 1 4 では、各分割ビット列をそれぞれエントリアドレスとしている。すなわち、図 1 9 に示した例では、合計 7 ビット分のビット列を、1 番上の階層では 2 ビット分、次の階層では 3 ビット分、一番下の階層では 2 ビット分の 3 階層の分割ビット列に分割している。そして、1 段目の検索テーブル 1 1 1 および 3 段目の検索テーブル 1 1 4 では、“0 0” ~ “1 1” の 4 つの分割ビット列をそれぞれエントリアドレスとしている。また、2 段目の検索テーブル 1 1 2、1 1 3 では、“0 0 0” ~ “1 1 1” の 8 つの分割ビット列をエントリアドレスとしている。

【 0 0 1 7 】

このプレフィックス拡張検索方法では、検索テーブルの登録アドレスを更新する際に、そのアドレスがあるテーブルの位置で終了するか次のテーブルに継続するかを表わすための終了・継続フラグ、次の転送先の IP アドレスなどを示す次テーブルポインタや送信インタフェースの情報そのものを、その検索テーブルの各エントリの中の、登録し得るすべてのエントリにそれぞれ重複して設定している。また、IP アドレスの登録にあたっては、各検索テーブルにおいて、各分割ビット列のビット長に満たないエントリのビット列を各分割ビット列の末端まで拡張している。たとえばある階層での分割ビットが 3 ビットであるにもかかわらず、エントリのビット列が 1 ビットあるいは 2 ビットである場合には、これらをそれぞれ 3 ビットに拡張している。このようにビット列を共通の長さのビット列にまで拡張すると、拡張前のビット列を共有する複数のエントリが、共通の登録 IP アドレスに対応することになる。

【 0 0 1 8 】

図 2 0 は、図 1 7 に示した元となる (Original) IP アドレスと 3 つのレベルに拡張された (Expanded) IP アドレスを対比して示したものである。この図 2 0 に示した例では、IP アドレス $P_1 \sim P_8$ の登録 IP アドレスをそれぞれ、各分割ビット列の最下位ビットまで拡張するようにしている。例えば、有効ビット列長としてのプレフィックス長が “1” である IP アドレス P_5 のビット列 “0 *” (図 2 0 (Original)) は、1 ビット分だけ拡張されて、2 ビットのビット列 “0 0” および “0 1” となる (図 2 0 (Expanded))。そして、1 段目の検索テーブル 1 1 1 (図 1 9) では、エントリアドレスがビット列 “0 0” および “0 1” の両方のエントリに、IP アドレス P_5 がそれぞれ重複する形で登録される。

【 0 0 1 9 】

また、例えば、プレフィックス長が “3” である IP アドレス P_2 のビット列 “1 1 1 *” (図 2 0 (Original)) は、2 ビット分だけ拡張されて、5 ビットのビット列 “1 1 1 0 0”、“1 1 1 0 1”、“1 1 1 1 0” および “1 1 1 1 1” の 4 つのエントリとなる (図 2 0 (Expanded))。2 段目の検索テーブル 1

1 2、1 1 3（図 1 9）では、3 ビット目～5 ビット目として得られた分割ビット列がエントリアドレスとなる。したがって、エントリアドレスが、“1 0 0”、“1 0 1”、“1 1 0” および “1 1 1” の 4 つのエントリに、I P アドレス P_2 がそれぞれ登録されることになる。更に、他の登録 I P アドレスも同様に拡張して登録される。

【0 0 2 0】

このように、ビット列を拡張して登録することにより、登録 I P アドレスを検索テーブルの特定のエントリアドレスと対応付けることができる。その結果、各段の検索テーブルの検索にあたり、登録 I P アドレスの分割ビット列をそのままフルに対応付けて一致検索処理を行うことができる。したがって、プレフィックス拡張検索方法を採用すると、検索回数は、最大でも、検索テーブルの段数（階層数）で済むことになり、I P アドレスの構成ビット数を最大検索回数とする先の例よりも検索回数を大幅に減少させることができる。例えば、図 1 9 に示した例では、3 段の検索テーブル（1 1 1）、（1 1 2、1 1 3）、（1 1 4）を設けているので、検索回数は最大でも 3 回で済む。このように、プレフィックス拡張検索方法によれば、検索回数を大幅に減らすことができる。

【0 0 2 1】

このプレフィックス拡張検索方法を使用したアドレス検索装置の検索テーブル 1 1 1～1 1 4 には、先に説明した終了・継続フラグの他に、出力先振り分け情報、出力先振り分け情報のポインタが有効か無効かを示すフラグ、次に検索するビット列の検索テーブルへのポインタ等のエントリ情報が設定されている。これらの情報は、登録可能なすべてのエントリに書きこまれる。たとえばビット列“0 *”の場合には、1 ビット分だけ拡張された後の 2 ビットのビット列“0 0”および“0 1”の双方についてエントリ情報が設定されることになる。

【0 0 2 2】

また、分割ビット列のビット長に満たないビット列を登録するときは、ビット列を末端まで拡張して登録することは先に説明した。ビット列を拡張すると、拡張前のビット列で包含されるすべての分割ビット列のテーブルに、拡張前のビット列に対応するエントリ情報をすべて書き込む。ただし、ここで書き込みについ

て問題が生じる。ビット列を拡張して登録する場合には、複数のビット列がそれぞれを拡張した結果として1つの拡張後のビット列に共通して対応してしまう場合があるからである。これについては次に説明するが、拡張前のビット列におけるその中の1つが拡張後のビット列と対応するような選別が行われる。したがって、このような場合にはその選別されたビット列に対応させてエントリ情報を書き込むことになる。

【 0 0 2 3 】

図 2 1 および図 2 2 は、拡張前のあるビット列の長さよりも長い他の拡張前のビット列が存在する場合の例を説明するためのものである。ここで図 2 1 は先の例とは異なる4つのIPアドレス $P_1 \sim P_4$ のビットパターンを表わしている。ここで例えばIPアドレス P_1 はビット列“0”（1ビット長）であり、これを4ビットのビット列に拡張するとするとビット例“0***”となる。具体的には“0000”、“0001”、“0010”、“0011”、“0100”、“0101”、“0110”、“0111”の8種類となる。

【 0 0 2 4 】

IPアドレス P_2 はビット列“010”（3ビット長）であり、これを4ビットのビット列に拡張するとするとビット例“010*”となる。具体的には“0100”、“0101”の2種類となる。IPアドレス P_3 はビット列“1010”（4ビット長）であるので、このままで4ビットのビット列となる。ビット列の拡張は必要ない。

【 0 0 2 5 】

IPアドレス P_4 はビット列“10”（2ビット長）であり、これを4ビットのビット列に拡張するとするとビット例“10**”となる。具体的には“1000”、“1001”、“1010”、“1011”の4種類となる。

【 0 0 2 6 】

図 2 2 は、拡張後の4ビットのビット列と、拡張されたIPアドレスを対応付けたものである。4ビットのビット列の組み合わせはこの図に示すように16通り存在する。ところが、図 2 1 に示した4つのIPアドレス $P_1 \sim P_4$ のビット列からは生じえないビット列が図 2 2 には含まれている。すなわち、第1のビット

が“1”で第2のビットも“1”のものは図21では存在しない。対象とするルータではそのようなビット列のルート情報が到来することが想定されないからである。そこで、図22では4つのビット列“1100”、“1101”、“1110”、“1111”はIPアドレスとして登録しない(NULL)。このため、残りの12通りのビット列をIPアドレス $P_1 \sim P_4$ と対応付けて登録することになる。たとえばビット列“0000”はIPアドレス P_1 として登録することになる。

【0027】

ここで、拡張後の複数のIPアドレスが4ビットのビット列に重複して対応する場合がある。そこで、これらのいずれを対応付けるかが問題となる。たとえばビット列“0100”は、元のビット列“0”のIPアドレス P_1 と、元のビット列“010”のIPアドレス P_2 と競合することになる。同様にビット列“0101”についてもIPアドレス P_1 およびIPアドレス P_2 の間で、いずれを対応付けるかが問題となる。

【0028】

このように拡張後の複数のIPアドレスが4ビットの所定のビット列に重複して対応する場合には、それぞれの拡張前のビット列の長い方を優先させて対応させる。すなわち長い方の拡張前のビット列に対応するエントリ情報を書き込むようにしている。これら2つの例の場合には、元のビット列“0”のIPアドレス P_1 よりも元のビット列“010”のIPアドレス P_2 の方が拡張前のビット列が長い。そこで、図22に示すようにこれらについてはIPアドレス P_2 が対応付けられることになる。ビット列“1010”についてのIPアドレス P_3 とIPアドレス P_4 の場合にも、同様の約束でIPアドレス P_3 が採択される。

【0029】

図23は、従来の検索テーブルの内容を表わしたものである。出力先振り分け情報を「NHP情報」と表示すると、この図に示す従来のプレフィックス拡張方法では、拡張したビット列“000”～“111”とこれら「NHP情報」を対応付けると共に、それぞれのビット列“000”～“111”に対応させて次の階層の検索テーブルのノードへのポインタを用意していた。ここで、「次の階層

のノードへのポインタ」を「L P T R」と略称することにする。

【 0 0 3 0 】

図 2 4 は、図 2 3 に対応するもので、従来における分割ビット列を管理するアドレス検索装置の構成の概要を表わしたものである。従来のアドレス検索装置 1 2 0 は、検索テーブル 1 2 1 とビット列管理テーブル 1 2 2 とが I P アドレス管理処理ブロック 1 2 3 と接続された構成となっている。このように従来のアドレス検索装置 1 2 0 は、ビット列管理テーブル 1 2 2 が必須のものとなっている。

【 0 0 3 1 】

図 2 5 は、図 2 4 に示した従来のアドレス検索装置で検索テーブルにおけるビット列の登録、削除あるいは更新を行う際のビット更新の処理の流れを表わしたものである。この場合、まず、対象となるビット列をビット列管理テーブル 1 2 2（図 2 4）に登録したり、ビット列管理テーブル 1 2 2 から削除したり、あるいは更新処理を行う（ステップ S 2 0 1）。次に、ビット列管理テーブル 1 2 2 から拡張前のビット列を取得する（ステップ S 2 0 2）。そして、先に説明したような処理でこれらのビット列を所定のビット数の列に拡張する（ステップ S 2 0 3）。次に、検索テーブル 1 2 1 に書き込むビット列を計算する（ステップ S 2 0 4）。この計算に際して、ビット列が同一のものがあつた場合には、図 2 1 で説明したように拡張前のビット列が長い方を選択する。以上のようにして求めたビット列を検索テーブル 1 2 1（図 2 4）に書き込んで（ステップ S 2 0 5）、更新が終了する（エンド）。

【 0 0 3 2 】

【発明が解決しようとする課題】

このように図 2 4 に示した従来のアドレス検索装置 1 2 0 では、検索テーブル 1 2 1 の他にビット列を管理するビット列管理テーブル 1 2 2 を持ち、ビット列を検索テーブル 1 2 1 に対して登録あるいは削除、更新を行うたびに分割ビット列を所定のビット数まで拡張する必要があつた。

【 0 0 3 3 】

またプレフィックス拡張検索方法では、以上説明したようにエントリ情報の書き込み規則が必要なため、検索テーブルに拡張前のビット列がどのような構成と

なっていたかを示す拡張前の情報を残すことができない。このため、ビット列の登録や削除を行う際に、拡張前のビット列の情報が必要になり、そのために検索テーブルを管理するためのテーブルが余計に必要なことになる。例えば、図 2 1 において、IP アドレス P_2 を削除するものとする。この場合、対応する 2 つのビット列 “0 1 0 0” と “0 1 0 1” を削除することになるが、この検索テーブルの 2 つのビット列 “0 1 0 0”、“0 1 0 1” の位置に何を書き込めばよいのか、書き残せばよいのかの情報は、このテーブル自体では判断できない。たとえば A というビット構成と B というビット構成が存在したとして、A の方がビット構成が長いのでこれが検索テーブルに登録されていたものとする。ところが、通信ネットワークの変更によってあるルータでは A の方が必要でなくなったものとする。このとき、A というビット構成とこれについての出力先振り分け情報を検索テーブルから削除することになる。ところが、この削除を行う際に、A というビット構成に対してどのビット構成が競合したかを示す情報が検索テーブルに登録されていない。したがって、A というビット構成を削除したときにどのビット列を代わりに登録するかを検索テーブル自体で判断することができない。

【0 0 3 4】

また、プレフィックス拡張検索方法を使用すると、登録したビット列に一致する数だけの出力先振り分け情報を検索テーブルに記録しておく必要がある。これらの出力先振り分け情報は、分割した分割ビット列を検索する各階層ごとに、保持しておく必要がある。このため、この方法を使用したアドレス検索装置は、保持するメモリサイズが大きくなるといった問題点があった。

【0 0 3 5】

そこで本発明の目的は、通信ネットワークのルーティングを行うための検索テーブルへのアドレスの追加や削除を容易にし、しかも検索テーブルを管理するためのメモリを小型化することのできるアドレス検索装置を提供することにある。

【0 0 3 6】

【課題を解決するための手段】

請求項 1 記載の発明では、(イ) 通信ネットワークから次のルートに振り分けるべきデータ信号を入力し、そのデータ信号の最終的な宛先に至るルートを表わ

したビット列からなるルート情報を予め定めたビット列ずつ先頭から区切ってそれぞれの階層の分割ビット列として順に取り出す分割ビット列取り出し手段と、

(ロ) 各ルート情報に対応したデータ信号の振り分け先を示す出力先振り分け情報を記憶した出力先振り分け情報記憶手段と、(ハ) 分割ビット列取り出し手段によって取り出す可能性のある分割ビット列をそれぞれ対応する階層に配置したときに形成される木構造の各節に相当する位置にそれぞれ検索テーブルノードを配置し、これらの検索テーブルノードの各々に、対応する分割ビット列から次の検索テーブルノードを指示する次検索テーブルノード指示情報あるいは次の検索テーブルノードが存在しないときには出力先振り分け情報記憶手段におけるデータ信号の振り分け先を示す出力先振り分け情報が記載されている出力先振り分け情報記憶手段の該当箇所を指示する出力先振り分け情報格納箇所指示情報を記憶した検索テーブルと、(ニ) 分割ビット列取り出し手段によって取り出した分割ビット列を用いて、第1の階層の検索テーブルノードから次検索テーブルノード指示情報が読み出される間、分割ビット列を順に取り出しながら検索テーブルノードを1階層ずつ階層を高めて検索を行い、出力先振り分け情報格納箇所指示情報が検索された時点でこの階層の移動を停止して検索する階層を確定する階層確定手段と、(ホ) この階層確定手段によって確定した検索テーブルノードから読み出された出力先振り分け情報格納箇所指示情報を用いて出力先振り分け情報記憶手段からデータ信号についての出力先振り分け情報を読み出す出力先振り分け情報読み出し手段とをアドレス検索装置に具備させる。

【0037】

すなわち請求項1記載の発明では、IPアドレス等のルーティングのためのルート情報を、分割ビット列取り出し手段が予め定めたビット列ずつ先頭から区切ってそれぞれの階層の分割ビット列として順に取り出すようにしている。そして、ルーティングの際のデータ信号の出力先の振り分けのための出力先振り分け情報を得ることができるまで先頭の分割ビット列から順に分割ビット列を使用するようにしている。本発明で使用する検索テーブルは、分割ビット列取り出し手段によって取り出す可能性のある分割ビット列をそれぞれ対応する階層に配置したときに形成される木構造の各節に相当する位置にそれぞれ検索テーブルノードを

配置した構成となっている。ここで、これらの検索テーブルノードの各々には、対応する分割ビット列から次の検索テーブルノードを指示する次検索テーブルノード指示情報あるいは次の検索テーブルノードが存在しないときには出力先振り分け情報記憶手段におけるデータ信号の振り分け先を示す出力先振り分け情報が記載されている出力先振り分け情報記憶手段の該当箇所を指示する出力先振り分け情報格納箇所指示情報を記憶している。そして、分割ビット列取り出し手段によって取り出した分割ビット列を用いて、第1の階層の検索テーブルノードから次検索テーブルノード指示情報が読み出される間、分割ビット列を順に取り出しながら検索テーブルノードを1階層ずつ階層を高めて検索を行い、出力先振り分け情報格納箇所指示情報が検索された時点でこの階層の移動を停止して、検索する階層を確定するようにしている。確定した階層の該当する検索テーブルノードから読み出される出力先振り分け情報格納箇所指示情報を用いて出力先振り分け情報記憶手段からデータ信号についての出力先振り分け情報を読み出すことで、そのデータ信号を出力する先を決定することができる。

【 0 0 3 8 】

このように請求項1記載の発明では、アドレス検索装置に次々入力されるルート情報を、各分割ビット列をそれぞれ枝とする木構造に沿って対応付けていき、他のルート情報と共通の枝（階層）となっている状態では、1つずつ階層を上げて行って次の分割ビット列を追加的に使用するようにしている。このようにして分割ビット列を次々と使用して階層を上げていくと、遂にはそのアドレス検索装置が振り分け先を確定できるだけのルートに関する情報が得られる。このときの検索テーブルノードに示している出力先振り分け情報格納箇所指示情報を用いて出力先振り分け情報記憶手段からデータ信号についての出力先振り分け情報を読み出すようにすることで、出力先振り分け情報記憶手段という1つのまとまった記憶手段から、ルート情報に対応した出力先振り分け情報を読み出して、データ信号の出力先の振り分けに用いることができる。本発明では木構造の各節に対応する位置に検索テーブルノードを配置しているので、最大でも階層の数だけの検索テーブルノードの検索を行えば足り、ルート情報を分割して得られた分割ビット列の総数に相当する検索テーブルノードの検索は必要としないので、処理が迅

速である。また、それぞれのルート情報に対応する出力先振り分け情報は、出力先振り分け情報記憶手段に一括して記憶しておくので、記憶領域の有効活用を行うことができるだけでなく、検索テーブルの変更や登録削除が容易になる。

【 0 0 3 9 】

請求項 2 記載の発明では、請求項 1 記載のアドレス検索装置で、検索テーブルノードのうちで次検索テーブルノード指示情報を格納しているもののみが、次検索テーブルノード指示情報を格納する領域を配置していることを特徴としている。

【 0 0 4 0 】

すなわち請求項 2 記載の発明では、木構造の枝のそれぞれ先端（最上層）の検索テーブルノードでは、次検索テーブルノード指示情報を格納する必要がないので、次検索テーブルノード指示情報を格納する領域をこれらについて省略することができ、この点でも記憶領域の有効活用を図ることができる。

【 0 0 4 1 】

請求項 3 記載の発明では、請求項 1 記載のアドレス検索装置で、各階層の検索テーブルノードは、そのテーブルノードに連結する次の階層の検索テーブルノードが存在するか否かを示す有効情報有無情報を配置していることを特徴としている。

【 0 0 4 2 】

すなわち請求項 3 記載の発明では、各階層の検索テーブルノードは、1 つ上の階層の検索テーブルノードが存在するかどうかを判別させるための有効情報有無情報を配置しており、これによって木構造のそれぞれの枝が先端であるかどうかを簡単に判別することができる。

【 0 0 4 3 】

請求項 4 記載の発明では、請求項 1 記載のアドレス検索装置で、出力先振り分け情報格納箇所指示情報は、出力先振り分け情報記憶手段における出力先振り分け情報が記憶されている領域と、その領域における出力先振り分け情報の格納箇所を示す番地とで構成されていることを特徴としている。

【 0 0 4 4 】

すなわち請求項4記載の発明では、出力先振り分け情報記憶手段から該当するルート情報の出力先振り分け情報を読み出す際に出力先振り分け情報記憶手段における出力先振り分け情報が記憶されている領域と、その領域における出力先振り分け情報の格納箇所を示す番地を用いてこの読み出しを行うようになっている。これは、あるルート情報に基づく分割ビット列を順次使用することで検索テーブルノードを上位の階層に向けて順次辿っていき、データ信号を振り分けるのに必要な情報が得られた時点の最終の検索テーブルノードが、出力先振り分け情報記憶手段における出力先振り分け情報が記憶されている複数の領域の1つを指示する場合には、この領域内におけるそのルート情報に対応した番地も最終の検索テーブルノードから読み出していくことで、目的とする出力先振り分け情報の格納箇所を特定することができるからである。

【0045】

請求項5記載の発明では、請求項4記載のアドレス検索装置で、出力先振り分け情報記憶手段における出力先振り分け情報が記憶されている領域は、対応する分割ビット列を構成するビット数の二乗の数の番地を備えていることを特徴としている。

【0046】

すなわち請求項5記載の発明では、出力先振り分け情報が記憶されている領域のそれぞれは対応する分割ビット列を構成するビット数の二乗の数の番地を備え、それぞれに出力先振り分け情報を登録できるようになっている。したがって、分割ビット列を構成するビット数を多くすると、それだけ該当領域が大きくなることになる。

【0047】

請求項6記載の発明では、請求項1記載のアドレス検索装置で、分割ビット列取り出し手段が順に取り出す分割ビット列は、それぞれ互いに等しいビット数で構成されていることを特徴としている。

【0048】

すなわち請求項6記載の発明では、分割ビット列取り出し手段が順に取り出す分割ビット列は、それぞれ互いに等しいビット数で構成されることが一例である

ことを示している。たとえばルート情報を3ビット単位で区切ってそれぞれを分割ビット列とすることができ、これにより分割処理を単純化すると共に前記した出力先振り分け情報が記憶されている領域のサイズの均一化を図ることができる。もちろん、分割ビット列は階層に応じて構成ビット数を変えるようにすることも自由である。

【0049】

【発明の実施の形態】

【0050】

【実施例】

以下実施例につき本発明を詳細に説明する。

【0051】

図1は本発明の一実施例のアドレス検索装置で使用する検索テーブルの要部を表わしたものである。本実施例の階層別の検索テーブル300には、ビット列を拡張せずに拡張前のビット列を記録するフィールドとしての出力先振り分け情報テーブル301と、次の階層のノードへのポインタ302が配置されている。この図1では図23と同様に分割ビット列が3ビットの場合を示している。図1でVBITフィールド303は、出力先振り分け情報を表わした出力先振り分け情報テーブル301のエントリが有効か否かを示すビット情報である。また、出力先振り分け情報テーブル301内の出力先振り分け情報 $NHP_1 \sim NHP_{14}$ は「次に転送するルータのIPアドレスまたは転送先のポート番号」、すなわちそのルータの出力の振り分け先（出力ポート）を示す情報を表わしている。

【0052】

本実施例の場合には、この図1にも示されているように検索テーブル300におけるビット列を拡張していない。その代わりに、拡張していないこのようなビット列に対応する出力先振り分け情報が検索テーブル300のどの位置に存在するかを示す情報を表わしたVBITフィールド303を持っている。VBITフィールド303の詳細については後に説明する。

【0053】

図2は、図25に対応するもので本実施例のアドレス検索装置の検索テーブル

でのビット更新の処理の様子を表わしたものである。この処理では、まず更新するビット列に対応する V B I T フィールド 3 0 3 (図 1) のビットを更新する (ステップ S 4 0 1)。次に、更新がルートの新設等に基づいた出力先振り分け情報 (N H P 情報) の登録の場合には、登録する拡張していないビット列に対応する出力先振り分け情報を、検索テーブル 3 0 0 (図 1) に書き込んで (ステップ S 4 0 2)、ビット更新処理を終了させる (エンド)。

【 0 0 5 4 】

本実施例では、図 2 に示したように、更新する分割ビット列に対応する V B I T フィールド 3 0 3 の 1 ビットを更新し、検索テーブル 3 0 0 に出力先振り分け情報を書き込むだけでよい。また、同じくルートの変更等によって所定のビット列に対応する出力先振り分け情報を削除するときは V B I T フィールド 3 0 3 のそれに対応する 1 ビットをビット “0” に設定すればよい。ただし、この前提としてビット “0” を未登録を表わす状態のものとし、ビット “1” を登録を表わす状態のものとする。

【 0 0 5 5 】

図 3 は、分割ビット列のビット数が “3” の場合の V B I T フィールドを表わしたものである。V B I T フィールド 3 0 3 のビット列が 1 ビットから 3 ビットまでの各ビットの組み合わせで構成されているものとする。本実施例では各ビット列を拡張する処理を行わないので、ビット列が 1 ビット構成の場合はビット “0” とビット “1” の 2 種類である。また、2 ビット構成の場合には、“0 0”、“0 1”、“1 0”、“1 1” の 4 種類である。3 ビット構成の場合には “0 0 0”、“0 0 1”、“0 1 0”、“0 1 1”、“1 0 0”、“1 0 1”、“1 1 0”、“1 1 1” の 8 種類である。したがって、これらの合計の 1 4 種類のビットの組み合わせが存在することになり、それぞれのビット列に 1 ビットを割り振るものとする、V B I T フィールド 3 0 3 全体で 1 4 ビットが必要になる。

【 0 0 5 6 】

たとえば、このうちのビット列 “0 0” に対応する位置に出力先振り分け情報を登録したいときには、V B I T フィールド 3 0 3 の “0 0” のビット (図 3 で左から 3 番目のビット) を “1” にして、対応する検索テーブルの “0 0” のエ

ントリに出力先振り分け情報を書き込む。また、削除したいときは、V B I T の削除した分割ビット列に対応する V B I T フィールド 3 0 3 のビットを“0”にする。このような処理だけで階層別の検索テーブル 3 0 0 への登録や変更、削除が終了する。

【0057】

図4は、本実施例におけるアドレス検索装置の構成を示したもので、従来の図24に対応するものである。このアドレス検索装置310は、分割ビット列のすべての階層の検索テーブルを保持した検索テーブル311と、分割ビット列の各階層の検索処理を行う分割ビット列検索ブロック312と、分割ビット列検索ブロック312で得られた最終的な検索結果情報313を基にして出力先振り分け情報を取得する出力先振り分け情報取得ブロック314と、出力先振り分け情報取得ブロック314が参照する出力先振り分け情報テーブル315とで構成されている。アドレス検索装置310の分割ビット列検索ブロック312にはI P アドレス316が供給され、出力先振り分け情報取得ブロック314は出力先振り分け情報テーブル315を検索して出力先振り分け情報317を出力するようになっている。ここで、分割ビット列検索ブロック312から出力される検索結果情報313は、ビット列E B I T、ポインタN N - P T R、S T A G E、および入力I P アドレスから構成されている。これらについては後に説明する。

【0058】

すなわち、図4に示す本実施例のアドレス検索装置310は、図24に示した従来のアドレス検索装置120と対比すると、検索テーブル121（図24）を検索テーブル311と出力先振り分け情報テーブル315に分け、分割ビット列を検索する各階層では検索テーブル311をアクセスし、全階層の検索が終了したあとに出力先振り分け情報テーブル315をアクセスするといった手法を採っている。これによって、従来の検索テーブル121で必要だった出力先振り分け情報テーブル315に対応するサイズを大幅に削除することができる。

【0059】

図5は図4に示した検索テーブルの1つのノードを示したものである。すなわち図4に示した検索テーブル311は、図5で示される検索テーブルノード32

1で構成されている。検索テーブルノード321は、分割ビット列をポインタを利用した木構造で管理するためのテーブル構成となっている。検索テーブルノード321は、次の階層の検索テーブルへのポインタLPTRが有効(Enable)か否かを識別するビット列EBIT(有効情報有無情報)と、出力先振り分け情報テーブル315ノードへのポインタNNPTRと、このノードにビット列が登録されている場合にどのようなビット列が登録されているかを示すビットVBITと、それぞれ次の階層の検索テーブルのノードへのポインタLPTR₁~LPTR_Mの各フィールドで構成されている。ここで、値Mは分割したビット列のビット数により異なり、例えば、3ビットで分割した前記した例の場合は、これらのビットの組み合わせの数として値Mは“8”となる。

【0060】

図6は、検索テーブルノードの構成例を示したものである。この図で検索テーブルノード321のビット列EBIT322は、分割ビット列のビット長が3ビットの場合を具体的に表わしたものであり、各ビットL₁~L₈はポインタLPTR₁~LPTR₈が有効であるか否かを示している。検索テーブルノード321のVBITフィールド303は、具体的にはビット“0”、“1”、“00”、…、“111”として記述してある。このビット“0”、“1”、“00”、…、“111”は、この検索テーブルノード321に分割ビット列“0”、“1”、“00”、“01”、…、“110”、“111”が登録されているか否かを識別するためのものである。たとえば、分割ビット列“11”(ビット長が“2”)が登録されているときは、VBITフィールド303におけるビット“11”のビットが“1”になり、ポインタNNPTRで示される出力先振り分け情報テーブル315のノードには、ビット列“11”に対応する出力先振り分け情報が格納されるようになっている。

【0061】

一方、図7は、図4に示した出力先振り分け情報テーブルの1つのノードを示したものである。出力先振り分け情報テーブルノード325には、分割ビット列の出力先振り分け情報が登録される。分割ビット列が3ビット構成の場合には、“0”、“1”、“00”、“01”、“10”、“11”、“000”、…、

“110”、“111”の14種類のビット列の出力先振り分け情報（ NHP_1 、 NHP_2 …… NHP_{14} ）が最大で登録されることになる。

【0062】

図4に戻って説明を続ける。出力先振り分け情報テーブル315は、図7で示される出力先振り分け情報テーブルノード325の複数個から構成されていて、検索テーブル311の中にあるポインタ $NNPTR$ により参照される。出力先振り分け情報テーブルノード325（図7）は、分割ビット列に対応する N 個の出力先振り分け情報 $NHP_1 \sim NHP_N$ で構成され、ビット拡張前のビット列すべてに対応する出力先振り分け情報を持つ。例えば、3ビットで分割したときは、ビット列“0”、“1”、“00”、“01”、“10”、“11”、“000”、…、“110”、“111”の14種類が考えられるので、パラメータ N は“14”になる。図7の出力先振り分け情報 NHP_N に対応するビット列が“xxxx”として表わされているが、これは分割したビット列のビット長により異なる。3ビットで分割したときはビット列“111”となり、4ビットで分割したときはビット列“1111”となる。

【0063】

分割ビット列検索ブロック312は、これに入力されるIPアドレス（以下入力IPアドレスという）316を、あらかじめ決められたビット長ごとに分割し、分割したビット列を用いてビット列EBITとポインタ $NNPTR$ の値を検索する処理を行うものである。

【0064】

出力先振り分け情報取得ブロック314は、分割ビット列検索ブロック312の処理で検索した、ビット列EBITとポインタ $NNPTR$ の値を最終的な検索結果情報313として受け取り、対応する出力先振り分け情報を出力する処理を行う。出力先振り分け情報テーブル315からポインタ $NNPTR$ をアドレスとするノードを計算し、ビット列EBITから最長一致する出力先振り分け情報317を、アドレス検索装置310から出力するようになっている。

【0065】

図4に示した本実施例のアドレス検索装置310ではIPアドレスのビット長

、分割数、分割ビット列を検索する階層数に制限は無いが、説明を簡単にするために I P アドレスのビット長を“9”、分割数を“3”ならびに分割ビット列の長さを“3”とした場合を次に具体的に説明する。この場合、図 5 に示した検索テーブルノード 3 2 1 のポインタ L P T R の数は、値 M が“8”となり、ポインタ L P T R₁ からポインタ L P T R₈ までの 8 個となる。

【 0 0 6 6 】

このようにポインタ L P T R が 8 個のため、ポインタ L P T R₁ ~ ポインタ L P T R₈ までのテーブルの有効あるいは無効を示すために必要なビット数としてのビット列 E B I T は“8”となる。ビット長が“3”であるときに考えられるすべてのビット列は“0”、“1”、“0 0”、“0 1”、“1 0”、“1 1”、“0 0 0”、…、“1 1 0”、“1 1 1”の 1 4 種類あるので、V B I T は 1 4 ビットになる。

【 0 0 6 7 】

次に、最長一致するプレフィックスを求めたい 9 ビットからなる I P アドレスが、図 4 に示したアドレス検索装置 3 1 0 の分割ビット列検索ブロック 3 1 2 に入力した後の動作を説明する。

【 0 0 6 8 】

図 8 は、検索したい I P アドレスを基にして、図 4 の出力先振り分け情報取得ブロックへ渡す検索結果情報を求める処理の流れを表わしたものである。具体的には、検索結果情報 3 1 3 (図 4 参照) として出力先振り分け情報取得ブロック 3 1 4 に渡される V B I T フィールド、ポインタ N N - P T R、S T A G E を保持しているパラメータとしての R E G _ V B I T、R E G _ N N - P T R、R E G _ S T A G E を求める処理である。本実施例では階層数を“3”としたので、変数 N は“1”から“3”まで採りうる。

【 0 0 6 9 】

入力 I P アドレス 3 1 6 が図 4 の分割ビット列検索ブロック 3 1 2 に入力されたものとする。分割ビット列検索ブロック 3 1 2 では、まず検索のための初期化処理を行う (図 8 ステップ S 4 2 1)。この初期化処理では、まず、処理を行う階層を表わすパラメータ n を“1”にする。また、検索テーブル 3 1 1 (図 4)

の次の階層への検索テーブルノードのポインタをポインタNEXTLPTRとし、「初期値」は検索テーブルの第1階層のノードへのポインタアドレスとする。第1階層は最初のビット列の検索なのでノードは1つしかない。

【0070】

ポインタREG_VBITとポインタREG_NN-PTRは、それぞれ最長一致した分割ビット列の検索テーブルノードのVBITフィールドと、出力先振り分け情報テーブルへのノード（以下出力先振り分け情報テーブルノードと呼ぶ。）へのポインタ情報を表わしている。パラメータREG_STAGEはポインタREG_VBITとポインタREG_NN-PTRを更新したときの階層を保持する変数である。図8に示す処理では、ポインタREG_VBITとポインタREG_NN-PTRを更新したときに、パラメータNが示す階層の値が代入される。パラメータREG_STAGEの初期値は、一致している検索テーブルノードを見つける前なので、無効とする。例えばプログラム言語等で実現するときは、負の値を入れるか、別の変数を用いて、ポインタREG_VBITとポインタREG_NN-PTR、およびパラメータREG_STAGEが無効であることを表わすことができる。

【0071】

終了フラグは、全階層の検索を終えたか、途中の階層でも次に検索すべき検索テーブルノードが無いときに終了フラグが“on（オン）”（有効）になる。初期値は“off（オフ）”である。

【0072】

初期化が終了すると、次に階層nの検索を行う（ステップS422）。初期化のすぐ後に、変数nは“1”になっている。このため、第1階層の検索テーブル311の処理を行う。

【0073】

図9は、ステップS422での検索テーブルの検索処理の流れを表わしたものである。まず、符号NEXTLPTRをアドレスとして1個の検索テーブルノードをリードする（S441）。次に、リードしたノードのVBITフィールドに“1”になるビットがあるか調べる（S442）。1つでも“1”になるビットが

存在した場合には、この検索テーブルノードに入力 I P アドレス 3 1 6 の分割ビット列が登録されている。そこでこの場合には、S 4 4 3 の手順を行う。V B I T フィールドの中に 1 つも “1” になっているビットが無いときには (N)、S 4 4 3 の処理を行わずに、ステップ S 4 4 4 の処理を行う。

【0 0 7 4】

ところで V B I T フィールドの中で 1 つでも “1” になっているビットがある場合には (ステップ S 4 4 2 : Y)、この検索テーブルノードに分割ビット列が登録されている。そこでポインタ R E G _ V B I T とポインタ R E G _ N N - P T R を、リードした検索テーブルノードの V B I T フィールドとポインタ N N - P T R に更新し、パラメータ R E G _ S T A G E をパラメータ n で書き込む。登録されている分割ビット列に対応する出力先振り分け情報テーブルノードは、検索テーブル 3 1 1 の中にはなく、出力先振り分け情報テーブル 3 1 5 の中に存在する。出力先振り分け情報テーブルノードへのポインタとしてポインタ N N - P T R が検索テーブルノードにある。したがって、分割ビット列検索ブロック 3 1 2 の処理が終了後、ポインタ R E G _ N N - P T R を利用し、出力先振り分け情報取得ブロック 3 1 4 で出力先振り分け情報 3 1 7 を取得する。

【0 0 7 5】

次に、検索している階層が最後の階層であるかどうかを判別する (ステップ S 4 4 4)。本実施例では階層数を “3” にしたので、検索している階層が “3” のときは次の階層が存在しない (Y)。そこでこの場合には、終了フラグを “o n” するために S 4 4 5 の手順を行う。

【0 0 7 6】

検索している階層が最後の階層ではない場合 (ステップ S 4 4 4 : N)、リードした検索テーブルノードの次の階層の検索テーブルノードの有無をビット列 E B I T のビットで調べる。そして、現在の階層で検索している入力 I P アドレスの分割ビット列、つまり、この例では 3 ビットから構成されるビット列で、その 3 ビットで表わされる “1” ~ “8” の数値に対応するポインタ L P T R₁ ~ L P T R₈ が有効であるか否かを、ビット列 E B I T のビット L₁ ~ L₈ (図 6 参照) から判別する (S 4 4 6)。有効 (“1”) であればステップ S 4 4 7 に進み

、無効（“0”）であればS 4 4 5の処理を行う。例えば、現在の階層で検索している入力IPアドレス3 1 6の分割ビット列が“0 1 0”であるとする。この場合、ビット列EBITの L_2 ビットが“1”になっていて有効であることを示していれば、次の階層の検索テーブルのノードが存在するので、ステップS 4 4 7の処理を行い、検索テーブルノードからポインタL PTR₂をリードし、次のステップS 4 4 8でアドレスNEXT L PTRをポインタL PTR₂で書き換える。

【0 0 7 7】

ビット列検索ブロックの処理が終了した後に、図4に示した出力先振り分け情報取得ブロック3 1 4で、検索したポインタREG_VBITとポインタREG_NN-PTR、パラメータREG_STAGEと入力IPアドレス3 1 6から出力先振り分け情報3 1 7を取得する。

【0 0 7 8】

図9に示した所定の階層についての検索処理が行われたら（図8ステップS 4 2 2）、すべての階層の検索処理が終了したかどうかの判別が行われる（ステップS 4 2 3）。そして、まだ検索が終了していない場合には（N）、次の階層の検索を行うためにパラメータ“n”を“1”だけ加算する（ステップS 4 2 4）。そして、ステップS 4 2 2に戻って、その加算後のパラメータ“n”で示される階層についての検索を行うことになる。

【0 0 7 9】

このようにしてすべての階層についての検索が終了したら（ステップS 4 2 3：Y）、入力IPアドレス3 1 6の分割ビット列検索処理が終了することになる（エンド）。

【0 0 8 0】

図10は、この出力先振り分け情報を取得する処理の流れを表わしたものである。まず、ポインタREG_VBITおよびポインタREG_NN-PTRが更新されたかどうかチェックされる（ステップS 4 7 1）。前段の分割ビット列検索ブロック3 1 2でポインタREG_VBITおよびポインタREG_NN-PTRが更新されなかったとき（N）、すなわち図9におけるステップS 4 4 3

で分割ビット列検索処理中に更新が一度も行われなかったとき、入力IPアドレス316に一致するビット列が登録されていないことになる。そこで、この場合には、エラーあるいはビット列一致無しの結果を出力する（ステップS472）。

【0081】

ポインタREG_VBITおよびポインタREG_NN-PTRが更新された場合には（ステップS471：Y）、更新したポインタREG_VBITには、分割ビット列検索ブロックの処理で最後に分割ビット列が一致した検索テーブルノードのVBITフィールドが格納されている。このVBITフィールドは図6のVBITフィールド303のように表わされる。入力IPアドレス316のビット列から階層REG_STAGEに相当するビット列を取り出し、取り出したビット列とVBITで表わされるビット列を比較し、より長いビット列を選択する（図10ステップS473）。

【0082】

そして、選択したビット列に対応する出力先振り分け情報317を、ポインタREG_NN-PTRと選択したビット列の位置から計算する。ポインタREG_NN-PTRは図7の出力先振り分け情報テーブルノード325の先頭を表わし、選択したビット列により出力先振り分け情報NHP₁～NHP₁₄のいずれかを選択する（図13のステップS474）。例えば、ビット列“010”を選択した場合は、図6よりビット列“010”は9番目に位置するので、出力先振り分け情報NHP₉を出力先振り分け情報317として取り出すことになる。

【0083】

次にIPアドレスのビット長を“9”、分割数を“3”ならびに分割ビット列の長さを“3”とした場合について、本実施例のアドレス検索装置310の動作を具体的に説明する。

【0084】

図11は、検索テーブルに登録されているIPアドレスを示したものである。図4に示した検索テーブル311には、IPアドレスP₁～P₅が登録されているものとする。IPアドレスP₁はビット長が“2”で、IPアドレスP₂のビット

長は“5”である。IPアドレス $P_1 \sim P_5$ に対応する出力先振り分け情報317を、出力先振り分け情報 $NHP(P_1)$ 、 $NHP(P_2)$ 、…、 $NHP(P_5)$ とする。これら5つのIPアドレス $P_1 \sim P_5$ を、検索テーブル311および出力先振り分け情報テーブル315に登録する。

【0085】

図12は、IPアドレス $P_1 \sim P_5$ に登録した場合の第1階層から第3階層までのIP検索テーブルを表わしたものであり、図13は同じくIPアドレス $P_1 \sim P_5$ に登録した場合の出力先振り分け情報テーブルを表わしたものである。図12に示す第1階層～第3階層のIP検索テーブル341～343のうち第3階層のIP検索テーブル343は分割したビット列の一番最後の階層のテーブルである。したがって、次の階層の検索テーブルノードへのポインタは不要である。このため、第1階層と第2階層のIP検索テーブル341、342で必要な検索テーブルノードにあるビット列EBITと出力先振り分け情報 $NHP_1 \sim NHP_8$ は、第3階層のIP検索テーブル343では不要のため、図示していない。

【0086】

IPアドレス P_1 のビット列“01”はビット長が“2”であるので、第1階層のIP検索テーブル341に登録される。図12の検索テーブルノード351のVBITフィールドの“01”に相当するビット（左から4番目のビット）が“1”になる。この検索テーブルノード351に対応する出力先振り分け情報テーブルノードはポインタNNPTRによって示されている。本例ではアドレス“5050(h)”を用い、図13の出力先振り分け情報テーブルノード361になる。また、出力先振り分け情報はビット列“01”のテーブル362（図13）に登録されている出力先振り分け情報 $NHP(P_1)$ になる。

【0087】

IPアドレス P_2 のビット列“00110”はビット長が“5”であるので、第2階層のIP検索テーブル342に登録される。第1階層のIP検索テーブル341には、第2階層の検索テーブルノードへのポインタを記録しておく必要がある。そこで、第1階層の検索テーブルノード351のビット列“001”に対応するテーブルに、第2階層の検索テーブルノードへのポインタとして“200

0 (h) ” を記録し、かつ、ビット列 “0 0 1” に対応するビット列 E B I T のビットを “1” にする。このポインタが示す検索テーブルノード 3 5 2 は図 1 2 のようになる。

【0 0 8 8】

I P アドレス P_2 の第 2 階層のビット列は、I P アドレス P_2 ビット列 “0 0 1 1 0” のうちの最初から 3 ビットまでのビット列 “0 0 1” を除いた 2 ビットのビット列 “1 0” となる。そこで、V B I T フィールドの “1 0” に相当するビット（左から 5 番目のビット）が “1” になる。このノードに対応する出力先振り分け情報テーブルノードはポインタ N N - P T R によって示されている。本例では “5 2 5 0 (h)” を用い、図 1 3 の出力先振り分け情報テーブルノード 3 7 1 になる。出力先振り分け情報はビット列 “1 0” のテーブル 3 7 2 (図 1 3) に登録されている出力先振り分け情報 N H P (P_2) になる。

【0 0 8 9】

I P アドレス P_3 のビット列 “0 0 1 1 0 1” はビット長が “6” であるので、同様に第 2 階層の I P 検索テーブルに登録される。第 1 階層の I P 検索テーブルには、第 2 階層の検索テーブルノードへのポインタを記録しておく必要があるが、I P アドレス P_2 と同じ分割ビット列を第 1 階層に持つ。したがって、第 1 階層テーブル 3 5 1 のビット列 “0 0 1” に対応するテーブルと、第 2 階層の検索テーブルノード 3 5 2 は I P アドレス P_2 と共有される。I P アドレス P_3 の第 2 階層のビット列は “1 0 1” となるので、V B I T フィールドの “1 0 1” (左から 1 2 番目のビット) が “1” になる (図 1 2 の検索テーブルノード 3 5 2) 。出力先振り分け情報テーブルノードは、I P アドレス P_2 と同じ場所に持つ。出力先振り分け情報はビット列 “1 0 1” のテーブル 3 7 3 (図 1 3) に登録されている出力先振り分け情報 N H P (P_3) になる。

【0 0 9 0】

I P アドレス P_4 と I P アドレス P_5 はそれぞれビット長が “8” であるので、第 3 階層の I P 検索テーブル 3 4 3 (図 1 2) に登録される。第 1 階層の I P 検索テーブル 3 4 1 には、第 2 階層の検索テーブルノードへのポインタを記録しておく必要がある。そこで、第 1 階層の検索テーブルノード 3 5 1 のビット列 “1

0 1” に対応するテーブルに、第 2 階層の I P 検索テーブル 3 4 2 のへのポインタとして “2 0 2 0 (h)” を記録し、かつ、ビット列 “1 0 1” に対応するビット列 E B I T のビットを “1” にする。このポインタが示す検索テーブルノード 3 5 3 は図 1 2 のようになる。

【0 0 9 1】

同様に、第 2 階層の I P 検索テーブル 3 4 2 に、第 3 階層の I P 検索テーブル 3 4 3 へのポインタを記録しておく必要がある。そこで、第 2 階層のビット列 “1 0 0” に対応するポインタに “3 0 3 0 (h)” を記録し、かつ、ビット列 “1 0 0” に対応するビット列 E B I T のビットを “1” にする。第 1 階層のビット列が “1 0 1” で、かつ、ビット長が “4” から “6” の I P アドレスは本例では登録していない（図 1 1 参照）。そこで V B I T フィールドのビットはすべて “0” になり、ポインタ N N - P T R はどの出力先振り分け情報テーブルノードも示さない。I P アドレス P_4 と I P アドレス P_5 に対応する第 3 階層の検索テーブルノード 3 5 4 は図 1 2 のようになる。本実施例の説明では、階層数を “3” としたので、第 3 階層までしか存在しない。そのため次の階層の検索テーブルノードへのポインタが不要であるので、第 3 階層の検索テーブルノード 3 5 4 ではビット列 E B I T、ポインタ L P T R₁ ~ L P T R₈ までのテーブルを削除することができる。

【0 0 9 2】

このように図 1 1 に示した 5 つの I P アドレスが検索テーブル 3 1 1 の第 1 ~ 第 3 の階層の I P 検索テーブル 3 4 1 ~ 3 4 3（図 1 2）に登録されているときに、図 4 に示す検索対象としての入力 I P アドレス 3 1 6 が入力されたものとする。このときの動作を次に説明する。

【0 0 9 3】

まず、入力 I P アドレス 3 1 6 がビット列 “0 0 1 1 0 1 1 1 1” であるときのアドレス検索装置 3 1 0 の動作を説明する。I P アドレス 3 1 6 が図 4 の分割ビット列検索ブロック 3 1 2 に入力されると、まず最初に図 8 で説明したステップ S 4 2 1 の処理で変数の初期設定が行われる。次に、アドレス N X T L P T R が示す検索テーブルノードをリードする。ここで、第 1 検索テーブルノードをア

ドレス“1000(h)”としたときは、アドレスNEXTLPTRの初期値は“1000(h)”である。そこで、“1000(h)”とする図12の検索テーブルノード351をリードする(図9ステップS441)。

【0094】

次に、入力IPアドレスの第1階層の長さが1、2、3のすべてのビット長のビット列“0”、“00”、“001”に、VBITフィールドが一致するかを調べる。VBITフィールドのビットは“0”、“1”、“00”、“01”、…、“111”に対応するビット列が存在するときに“1”になる。検索テーブルノード351には“0”、“00”、“001”に対応するビットが“1”になっているものがないので(ステップS442:N)、ステップS443をスキップしてステップS444に進む。

【0095】

現在、入力IPアドレス316の第1階層の分割ビット列の検索処理をしているので、最後の階層の検索処理ではない(パラメータnは“3”未満)(ステップS444:N)。そこでステップS446に進む。ステップS446では、分割ビット列“001”に対応するビット列EBITのビットを調べると“1”になっている。したがって、次の階層の検索テーブルノードへのポインタが存在する(ステップS446:Y)。そこでステップS447へ進む。ステップS447では、検索テーブルノードから、分割ビット列“001”に対応するポインタLPTRをリードする。この値は2000(h)であるので、アドレスNEXTLPTRを2000(h)に更新する(ステップS448)。以上の手順で第1階層の検索は終了になり、終了フラグが“off”のままであるので検索は終了しておらず(図8ステップS423:N)、ステップS424に進み、パラメータnに“1”を加算する。そして、パラメータを“2”に設定した状態でステップS422で再度処理を行うことになる。

【0096】

更新されたアドレスNEXTLPTRは“2000(h)”である。そこで、図12の検索テーブルノード352をリードする(図9ステップS441)。入力IPアドレス316の第2階層の分割ビット列のすべての長さのビット列“1”

、“10”、“101”に一致するVBITフィールドが示すビットが“1”になっているビットが存在する（ステップS442）。そこで、リードしたポインタNN-PTRとVBITフィールドをそれぞれポインタREG_NN-PTRとポインタREG_VBITで書き込み、パラメータREG_STAGEにパラメータ“N”を書き込む（ステップS443）。パラメータ“n”は“2”であるのでステップS444では最後の階層とはされない（N）。そこで、ステップS446に進む。ステップS446で、入力IPアドレス316の第2階層の分割ビット列“101”に対応するビット列EBITのビットは“0”になっている。したがって、次の階層の検索ノードを示すポインタLPTRは1つも無い（N）。そこでステップS445に進み、終了フラグを“on”にする。図8のステップS423で、終了フラグの比較が行われ、“on”になっているので（Y）、入力IPアドレス316の分割ビット列検索処理は終了する（エンド）。

【0097】

このようにして分割ビット列検索処理が終了すると、図4に示されている分割ビット列検索ブロック312は、ポインタREG_VBITとポインタREG_NN-PTR、パラメータREG_STAGEおよび入力IPアドレス316を、検索結果情報313として出力先振り分け情報取得ブロック314に渡す。

【0098】

出力先振り分け情報取得ブロック314の処理を先に示した図10を用いて説明する。まず、ポインタREG_VBITおよびポインタREG_NN-PTRについて見てみる。図9のステップS443で、ポインタREG_VBITは“00001000000100”で、ポインタREG_NN-PTRは“5250(h)”と更新されており、有効となっている（図10ステップS471）。したがって、次のステップS473に処理が進む。ステップS473では、パラメータREG_STAGEは“2”である。そこで、第2階層の入力IPアドレスビット列“101”から、これに一致するポインタREG_VBITのVBITフィールドが示すビット列を調べる。これにより、ビット列“10”とビット列“101”が存在することが分かる。最も長く一致するビット列を選択するので、ビット列“101”を選択する（ステップS473）。ポインタREG_N

N-PTRは“5250(h)”であるので図13の出力先振り分け情報テーブルノード371を指している。ポインタREG_VBIT列と入力IPアドレスの比較により、一致する分割ビット列は“101”となる。したがって、ステップS474では、“5250(h)”で指される出力先振り分け情報テーブルノード373(図13)のビット列“101”の出力先振り分け情報をリードする。この場合、出力先振り分け情報は出力先振り分け情報NHP(P₃)となる。

【0099】

図14は、図12および図13の検索テーブルおよび出力先振り分け情報テーブルに、ビット列“10101”を登録するときの処理を説明するためのものである。テーブルを更新する場合はVBITフィールドの該当するビットの状態を反転させて、削除(“0”にする。)あるいは登録(“1”にする。)の操作を行うことができる。図12の第2階層の検索テーブルノード352には、ビット列“10”(図12で左から5番目のビットが“1”。)とビット列“101”(同図で右から3番目のビットが“1”。)が登録されている。この検索テーブルノード352に新たにビット列“01”を登録するものとする。このときは、まず図14の第2階層の検索テーブルノード364における、ビット列“01”に対応するVBITフィールド303のビットの位置365(左から4番目のビット)を“1”にする。次に、出力先振り分け情報テーブル315の検索テーブルノード366(図14)のビット列“01”に出力先振り分け情報NHP*を書き込む。図14では、図12および図13で表わされるIP検索テーブルに、ビット列“01”とそれに対応する出力先振り分け情報NHP*を登録した後の状態を示している。

【0100】

以上説明したように、従来の検索テーブル更新処理では、展開したビット列を検索テーブルに書き込むようにしているので、この検索テーブルには展開前のビット列の情報は残らない。このため、図24のビット列管理テーブル122の展開前のビット列を含んだ検索テーブル121を保持しておかなければいけない。この従来のアドレス検索装置120では、検索テーブル121の更新を行う場合には図25に示した処理を実施しなければならず、更新手順が複雑化し、更新の

ための検索テーブルを多く実装しておかなければいけない。

【0101】

これに対して図4に示した本実施例のアドレス検索装置310では、図2の処理で示すように、展開前のビット列に対応するVBITフィールド303および出力先振り分け情報を書き換えればよく、更新処理の手順が少なくなり、かつ、検索テーブル311以外のテーブルを別途持つ必要がなくなる。この結果、余計なメモリは必要なくなる。

【0102】

従来では、特開平11-191781号公報で開示しているように、次の階層のテーブルノードへのポインタと、出力先振り分け情報テーブルの両方を検索テーブルの全階層に必要としていた。これに対して本実施例の検索テーブルは、次の階層のテーブルノードへのポインタのみ持つテーブル（図5）と、出力先振り分け情報を持つテーブル（図7）に分けている。これにより、たとえば実施例で説明した図12における第2階層のIP検索テーブル342あるいは第3階層のIP検索テーブル343内に、図示していないがビット列が登録されない検索テーブルノードがあるような場合には、これらの検索テーブルノードを省略し、テーブル量を減らすことができる効果がある。

【0103】

本発明の第1の変形例

【0104】

先の実施例では検索テーブルは各階層の検索テーブルノードを束ねた形で1組で構成されていたが、これに限るものではない。検索テーブルを階層ごとに独立させることで、各階層の検索処理を並列に動作させ、検索性能を向上させることができる。

【0105】

図15は、本発明の第1の変形例におけるアドレス検索装置を示したものである。このアドレス検索装置310Aは、第1～第3階層分割ビット列検索ブロック312₁～312₃が縦続接続された構成となっており、第3階層分割ビット列検索ブロック312₃の後段に出力先振り分け情報取得ブロック314Aが配置

されている。第 1 階層分割ビット列検索ブロック 3 1 2₁は、第 1 階層検索テーブル 3 1 1₁を参照するようになっている。同様に第 2 階層分割ビット列検索ブロック 3 1 2₂は、第 2 階層検索テーブル 3 1 1₂を参照し、第 3 階層分割ビット列検索ブロック 3 1 2₃は、第 3 階層検索テーブル 3 1 1₃を参照するようになっている。入力 IP アドレス 3 1 6 は第 1 階層分割ビット列検索ブロック 3 1 2₁に入力されるようになっており、出力先振り分け情報取得ブロック 3 1 4 A は出力先振り分け情報テーブル 3 1 5 A を参照して出力先振り分け情報 3 1 7 を取得し、外部に出力するようになっている。

【 0 1 0 6 】

この第 1 の変形例のアドレス検索装置 3 1 0 A も、IP アドレスのビット長、分割する階層数、分割ビット列の長さに制限はないが、先の実施例と同様に、IP アドレスのビット長が“9”、分割数が“3”ならびに分割ビット列の長さが“3”の場合を示している。分割する階層数がこれ以外の値の場合には、これに応じて分割ビット列検索ブロック 3 1 2 および検索テーブル 3 1 1 の総数が加減されることになる。

【 0 1 0 7 】

この変形例のアドレス検索装置 3 1 0 A で第 1 階層分割ビット列検索ブロック 3 1 2₁は、図 8 に示したステップ S 4 2 1 の初期化手順で、パラメータ n を“1”としたときの、図 9 に示した第 1 階層の検索処理（ステップ S 4 4 1 ～ステップ S 4 4 8）を行う。このようにして第 1 階層分割ビット列検索ブロック 3 1 2₁の処理が終了すると、第 1 階層分割ビット列検索ブロック 3 1 2₁から第 2 階層分割ビット列検索ブロック 3 1 2₂に、検索後の終了フラグ、NEXTLPTR、ポインタ REG_VBIT、ポインタ REG_NN-PTR、パラメータ REG_STAGE ならびに IP アドレスビット列のパラメータからなる第 1 階層検索結果情報 3 1 3₁を引き渡す。

【 0 1 0 8 】

第 2 階層分割ビット列検索ブロック 3 1 2₂は、第 1 階層分割ビット列検索ブロック 3 1 2₁から引き渡された第 1 階層検索結果情報 3 1 3₁を用い、パラメータ n を“2”としたときの第 2 階層の検索処理を図 9 に示す各ステップ S 4 4 1

～S 4 4 8 で実行する。第 2 階層分割ビット列検索ブロック 3 1 2₂ も同様に、検索後の終了フラグ、NEXTLPTR、ポインタ REG_VBIT、ポインタ REG_NN-PTR、パラメータ REG_STAGE ならびに IP アドレスビット列のパラメータからなる第 2 階層検索結果情報 3 1 3₂ を第 3 階層分割ビット列検索ブロック 3 1 2₃ に引き渡す。ただし、第 1 階層分割ビット列検索ブロック 3 1 2₁ から引き渡された第 1 階層検索結果情報 3 1 3₁ の終了フラグが “on” になっているときには、図 9 に示した第 2 階層の検索処理（ステップ S 4 4 1 ～S 4 4 8）を行うことなく、引き渡されたパラメータとしての第 2 階層検索結果情報 3 1 3₂ を第 3 階層分割ビット列検索ブロック 3 1 2₃ にそのまま引き渡す。

【0 1 0 9】

第 3 階層分割ビット列検索ブロック 3 1 2₃ では、第 2 階層分割ビット列検索ブロック 3 1 2₂ から引き渡された第 2 階層検索結果情報 3 1 3₂ を用い、パラメータ n を “3” としたときの、第 3 階層の検索処理を図 9 に示す各ステップ S 4 4 1 ～S 4 4 8 で実行する。ステップ S 4 4 5 あるいはステップ S 4 4 8 の処理が終了したあとは、出力先振り分け情報取得ブロック 3 1 4 A に、検索後の終了フラグ、NEXTLPTR、ポインタ REG_VBIT、ポインタ REG_NN-PTR、パラメータ REG_STAGE ならびに IP アドレスビット列のパラメータからなる第 2 階層検索結果情報 3 1 3₂ を引き渡す。ただし、第 2 階層分割ビット列検索ブロック 3 1 2₃ から引き渡された第 2 階層検索結果情報 3 1 3₂ の終了フラグが “on” になっているときには、第 3 階層の検索処理（ステップ S 4 4 1 ～S 4 4 8）を行うことなく、引き渡されたパラメータとしての第 3 階層検索結果情報 3 1 3₃ を出力先振り分け情報取得ブロック 3 1 4 A にそのまま引き渡す。

【0 1 1 0】

この第 1 の変形例における出力先振り分け情報取得ブロック 3 1 4 A の処理は、先の実施例の出力先振り分け情報取得ブロック 3 1 4 の処理と同じである。そこでその処理は図 1 0 をそのまま用いることができる。

【0 1 1 1】

本発明の第2の変形例

【0112】

図16は本発明の第2の変形例におけるアドレス検索装置の出力先振り分け情報取得ブロックの要部を表わしたものである。この変形例では、図4に示した出力先振り分け情報取得ブロック314において、パラメータREG_STAGEと入力IPアドレス、ポインタREG_VBITから出力先振り分け情報317を求めるためのオフセットポインタを計算するようになっている。

【0113】

ここでオフセットポインタとは、出力先振り分け情報テーブルノード内の各ビット列の位置を表わすものである。たとえば、図13の出力先振り分け情報テーブルノード371におけるビット列“101”は、先頭のビット列“0”から数えて12番目のビット列である。そこでこの場合のオフセットポインタは“12”となる。

【0114】

第2の変形例のオフセットポインタ計算部500は、パラメータREG_STAGE501と入力IPアドレス502を入力するビット列取り出し手段503と、REG_VBIT504を入力するREG_VBIT分配手段505を備えている。ビット列取り出し手段503は、パラメータREG_STAGE501で表わされる階層に対応する入力IPアドレス502のビット列を取り出す。取り出されたパラメータ507は入力IPアドレス分配手段508に引き渡されるようになっている。入力IPアドレス分配手段508は入力IPアドレス509を第1～第Mのビット列サーチ手段 $511_1 \sim 511_M$ に分配する。

【0115】

一方、REG_VBIT分配手段505は、REG_VBIT504を各ビット長に分割して、それぞれのビット長ごとに第1～第Mのビット列サーチ手段 $511_1 \sim 511_M$ に引き渡すようになっている。第1～第Mのビット列サーチ手段 $511_1 \sim 511_M$ の後段には最長一致検索手段513が設けられており、入力されたオフセットポインタ $514_1 \sim 514_M$ を基にして、出力先振り分け情報へのオフセットポインタ515を出力するようになっている。

【0 1 1 6】

ここでビット列取り出し手段503は、パラメータREG_STAGE501で表わされる階層に対応する入力IPアドレスのビット列を取り出し、入力IPアドレス分配手段508にパラメータを引き渡す。先の実施例を例に採ると、入力IPアドレス502のビット列を3ビットずつに分割し階層数を“3”にし、各階層のビット長は“3”である。この場合、パラメータREG_STAGE501の値が“1”の場合には、第1階層であるのでビット“1”からビット“3”までを取り出して、入力IPアドレス分配手段508に引き渡す。パラメータREG_STAGE501の値が“2”の場合には第2階層であるので、ビット“4”からビット“6”までを取り出し、“3”の場合はビット“7”からビット“8”までを取り出し、それぞれ入力IPアドレス分配手段508に引き渡すことになる。

【0 1 1 7】

第1～第Mのビット列サーチ手段 $511_1 \sim 511_M$ は入力IPアドレス502のビット列の一部とREG_VBIT504を比較して、入力IPアドレス分配手段508から引き渡される入力IPアドレス502のビット列の一部がREG_VBIT504で表わされるビット列と一致しているかを調べる。そして、これらが一致している場合には、最長一致検索手段513にオフセットポインタ $514_1 \sim 514_M$ を引き渡す。分割した階層のビット長が“3”の場合には、値Mは“3”であり、第1～第3のビット列サーチ手段 $511_1 \sim 511_3$ が配置されている。これら第1～第3のビット列サーチ手段 $511_1 \sim 511_3$ は、特定のビット長のビット列を比較するために用いられる。

【0 1 1 8】

例えば、第1のビット列サーチ手段 511_1 は、入力IPアドレス502のビット列の一部であるビット長“1”のビット列と、ビット長が“1”であるビット列が登録されているか否かを示すREG_VBIT504の最初の2ビットを比較し、一致している場合は、出力先振り分け情報テーブルノードのビット列“0”あるいはビット列“1”のオフセットポインタ 514_1 を、最長一致検索手段513に引き渡す。

【 0 1 1 9 】

入力 I P アドレス分配手段 5 0 8 は、ビット列取り出し手段 5 0 3 から引き渡されたビット列を、第 1 のビット列サーチ手段 $5\ 1\ 1_1$ から第 M のビット列サーチ手段 $5\ 1\ 1_M$ までに分配する。例えば第 1 の実施例を例にすると、第 1 のビット列サーチ手段 $5\ 1\ 1_1$ には取り出したビット列の上位 1 ビットを渡す、第 2 のビット列サーチ手段 $5\ 1\ 1_2$ には取り出したビット列の上位 1 ビットから 2 ビット目までのビット列を渡す。第 M のビット列サーチ手段 $5\ 1\ 1_M$ には取り出したすべてのビット列を引き渡す。

【 0 1 2 0 】

REG_VBIT 分配手段 5 0 5 は、REG_VBIT 5 0 4 を各ビット長に分割して、それぞれのビット長ごとに第 1 ～第 M のビット列サーチ手段 $5\ 1\ 1_1 \sim 5\ 1\ 1_M$ へ引き渡す。第 1 のビット列サーチ手段 $5\ 1\ 1_1$ にはポインタ REG_VBIT の 1 ビット目から 2 ビット目の 2 ビットを引き渡す。第 2 のビット列サーチ手段 $5\ 1\ 1_2$ にはポインタ REG_VBIT の 3 ビット目から 6 ビット目までの 4 ビットを引き渡す。第 M のビット列サーチ手段 $5\ 1\ 1_M$ には、ポインタ REG_VBIT の右から 2 の M 乗ビットを、引き渡す。例えば、階層のビット長が“3”の場合は、値 M は“3”となる。このときの REG_VBIT 5 0 4 は図 3 に示された通りである。すなわち、最初の 2 ビットが 1 ビット長で、3 ビット目から 6 ビット目までが 2 ビット長、7 ビット目から 1 4 ビット目までが 3 ビットとなる。値 M が“3”のこの場合、これらのビット列がそれぞれ第 1 ～第 3 のビット列サーチ手段 $5\ 1\ 1_1 \sim 5\ 1\ 1_3$ へ引き渡される。

【 0 1 2 1 】

最長一致検索手段 5 1 3 は、第 1 ～第 M のビット列サーチ手段 $5\ 1\ 1_1 \sim 5\ 1\ 1_M$ からそれぞれ引き渡されるオフセットポインタ $5\ 1\ 4_1 \sim 5\ 1\ 4_M$ のうち、ビット長が最長のビット列サーチ手段（これをビット列サーチ手段 $5\ 1\ 1_X$ とする。）のオフセットポインタ $5\ 1\ 4_X$ を採用する。そして、出力先振り分け情報 3 1 7（図 4）へのオフセットポインタとして取り出す。例えば、第 2 のビット列サーチ手段 $5\ 1\ 1_2$ と第 3 のビット列サーチ手段 $5\ 1\ 1_3$ の両方からオフセットポインタを渡されたときは、ビット長の長い第 3 のビット列サーチ手段 $5\ 1\ 1_3$ か

ら受け取ったオフセットポインタを採用することになる。

【 0 1 2 2 】

この変形例では、図 1 6 に示すような構成としたので、ビット列の比較を並行して処理することができる。このため、ハードウェアや並列プロセッシング可能な回路や装置でアドレス検索装置を実現することが容易になり、単位時間あたりの検索数の性能を向上させることができる。

【 0 1 2 3 】

なお、以上説明した実施例では分割ビット列を 3 ビットずつ等分割されたビット列として示したが、3 ビット以外のビット数で分割するものであってもよい。また、分割ビット列は互いに等しいビット列である必要はなく、異なったビット列構成となってもよい。

【 0 1 2 4 】

また、実施例では IP アドレスの検索について説明したが、これ以外に分岐した経路を順に辿っていくようなルート情報を有するすべての検索に本発明を適用することができることは当然である。

【 0 1 2 5 】

【発明の効果】

以上説明したように請求項 1 記載の発明によれば、アドレス検索装置に次々入力されるルート情報を、各分割ビット列をそれぞれ枝とする木構造に沿って対応付けていき、他のルート情報と共通の枝（階層）となっている状態では、1 つずつ階層を上げて行って次の分割ビット列を追加的に使用するようにすると共に、木構造の各節に対応する位置に検索テーブルノードを配置しているので、最大でも階層の数だけの検索テーブルノードの検索を行えば足り、ルート情報を分割して得られた分割ビット列の総数に相当する検索テーブルノードの検索は必要としないので、検索処理が迅速となる。また、それぞれのルート情報に対応する出力先振り分け情報は、出力先振り分け情報記憶手段に一括して記憶しておくので、記憶領域の有効活用を行うことができるだけでなく、検索テーブルの変更や登録削除が容易になるという利点がある。

【 0 1 2 6 】

また請求項 2 記載の発明によれば、検索テーブルノードのうちで次検索テーブルノード指示情報を格納しているもののみが、次検索テーブルノード指示情報を格納する領域を配置することができ、これにより検索テーブル全体のメモリ容量を削減することができる。

【 0 1 2 7 】

更に請求項 3 記載の発明によれば、各階層の検索テーブルノードは、1 つ上の階層の検索テーブルノードが存在するかどうかを判別させるための有効情報有無情報を配置しているので、これによって木構造のそれぞれの枝が先端であるかどうかを簡単に判別することができる。

【 0 1 2 8 】

更に請求項 6 記載の発明によれば、分割ビット列取り出し手段が順に取り出す分割ビット列は、それぞれ互いに等しいビット数で構成されるので、分割処理を単純化すると共に前記した出力先振り分け情報が記憶されている領域のサイズの均一化を図ることができる。

【図面の簡単な説明】

【図 1】

本発明の一実施例のアドレス検索装置で使用する検索テーブルの要部を表わした説明図である。

【図 2】

本実施例のアドレス検索装置の検索テーブルでのビット更新の処理の様子を表わした流れ図である。

【図 3】

本実施例で分割ビット列のビット数が“3”の場合の V B I T フィールドを表わした説明図である。

【図 4】

本実施例におけるアドレス検索装置の構成を示したブロック図である。

【図 5】

図 4 に示した検索テーブルの 1 つのノードを示した説明図である。

【図 6】

分割ビット列を 3 ビットとした場合の検索テーブルノードの構成例を示した説明図である。

【図 7】

図 4 に示した出力先振り分け情報テーブルの 1 つのノードを示した説明図である。

【図 8】

本実施例で出力先振り分け情報取得ブロックへ渡す検索結果を求める処理の流れを表わした流れ図である。

【図 9】

本実施例で検索テーブルの検索処理の流れを表わした流れ図である。

【図 1 0】

本実施例で出力先振り分け情報を取得する処理の流れを表わした流れ図である。

【図 1 1】

本実施例で検索テーブルに登録されている IP アドレスを示した説明図である。

【図 1 2】

本実施例で IP アドレス $P_1 \sim P_5$ を登録した場合の第 1 階層から第 3 階層までの IP 検索テーブルを表わした説明図である。

【図 1 3】

本実施例で IP アドレス $P_1 \sim P_5$ を登録した場合の出力先振り分け情報テーブルを表わした説明図である。

【図 1 4】

図 1 2 および図 1 3 の IP 検索テーブルおよび出力先振り分け情報テーブルに、ビット列 “1 0 1 0 1” を登録するときの処理を示した説明図である。

【図 1 5】

本発明の第 1 の変形例におけるアドレス検索装置を示したブロック図である。

【図 1 6】

本発明の第 2 の変形例におけるアドレス検索装置の出力先振り分け情報取得ブ

ロックの要部を表わしたブロック図である。

【図 1 7】

8つのIPアドレス $P_1 \sim P_8$ のビットパターンを表わした説明図である。

【図 1 8】

従来のバイナリ・ツリー検索法を示した説明図である。

【図 1 9】

従来のプレフィックス拡張検索方法を示した説明図である。

【図 2 0】

図 1 7 に示した元となる IP アドレスと 3 つのレベルに拡張された IP アドレスを対比して示した説明図である。

【図 2 1】

4つのIPアドレス $P_1 \sim P_4$ のビットパターンを表わした説明図である。

【図 2 2】

拡張後の 4 ビットのビット列とこれに拡張された IP アドレスを対比した説明図である。

【図 2 3】

従来の検索テーブルの内容を表わした説明図である。

【図 2 4】

従来における分割ビット列を管理するアドレス検索装置の構成の概要を表わしたブロック図である。

【図 2 5】

従来のアドレス検索装置のビット更新の処理の流れを表わした流れ図である。

【符号の説明】

3 0 0 検索テーブル

3 0 1、3 1 5 出力先振り分け情報テーブル

3 0 2 ポインタ

3 0 3 V B I T フィールド

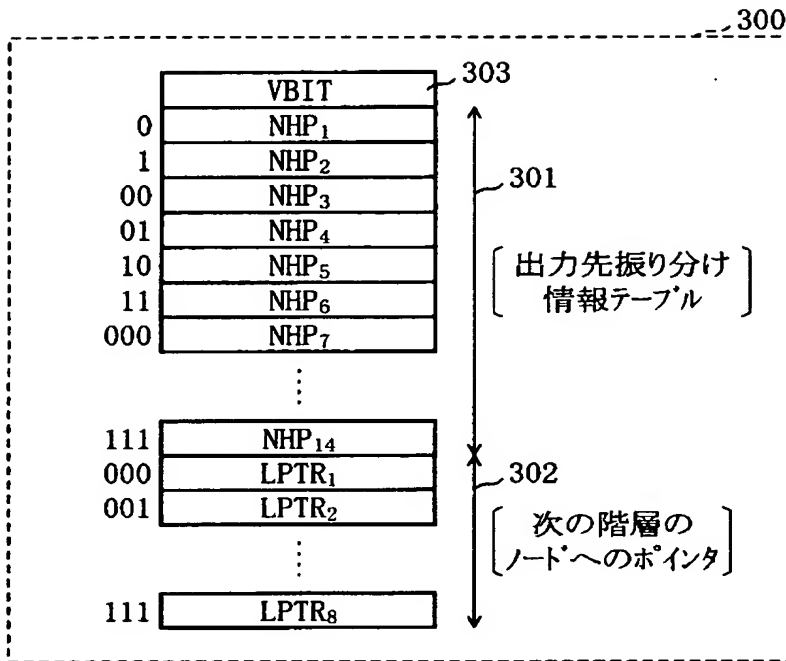
3 1 0、3 1 0 A アドレス検索装置

3 1 1 検索テーブル

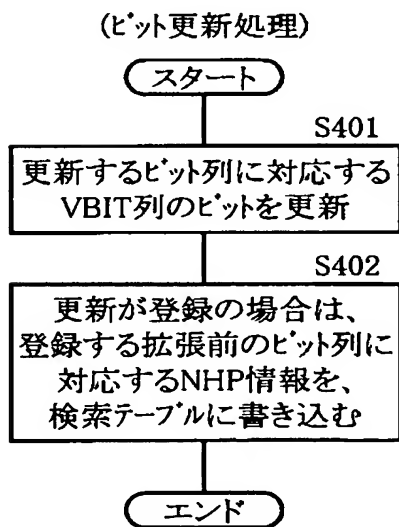
- 3 1 2 分割ビット列検索ブロック
- 3 1 4、3 1 4 A 出力先振り分け情報取得ブロック
- 3 1 6 I P アドレス
- 3 2 1、3 5 1 ～ 3 5 4 検索テーブルノード
- 3 2 2 ビット列 E B I T
- 3 4 1 第 1 階層の I P 検索テーブル
- 3 4 2 第 2 階層の I P 検索テーブル
- 3 4 3 第 3 階層の I P 検索テーブル
- 5 0 8 入力 I P アドレス分配手段
- 5 1 3 最長一致検索手段

【書類名】 図面

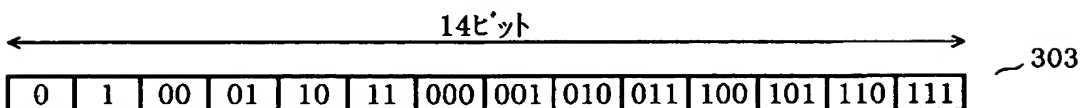
【図 1】



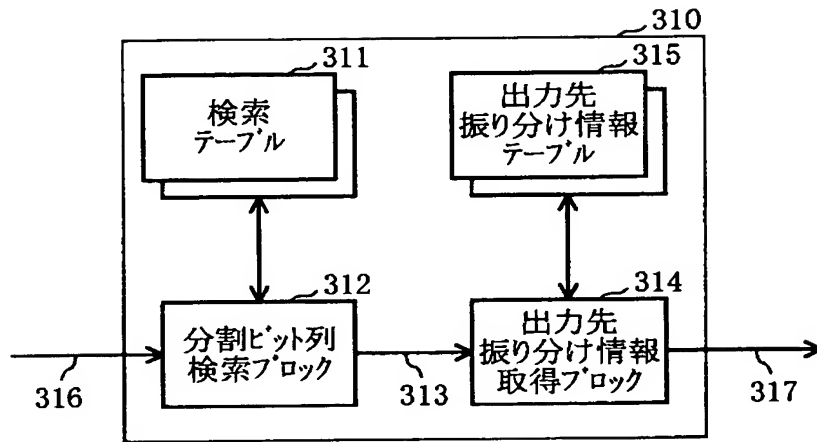
【図 2】



【図 3】

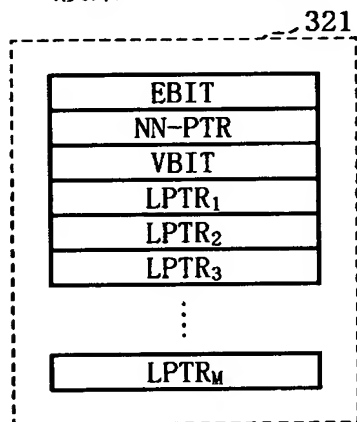


【図 4】

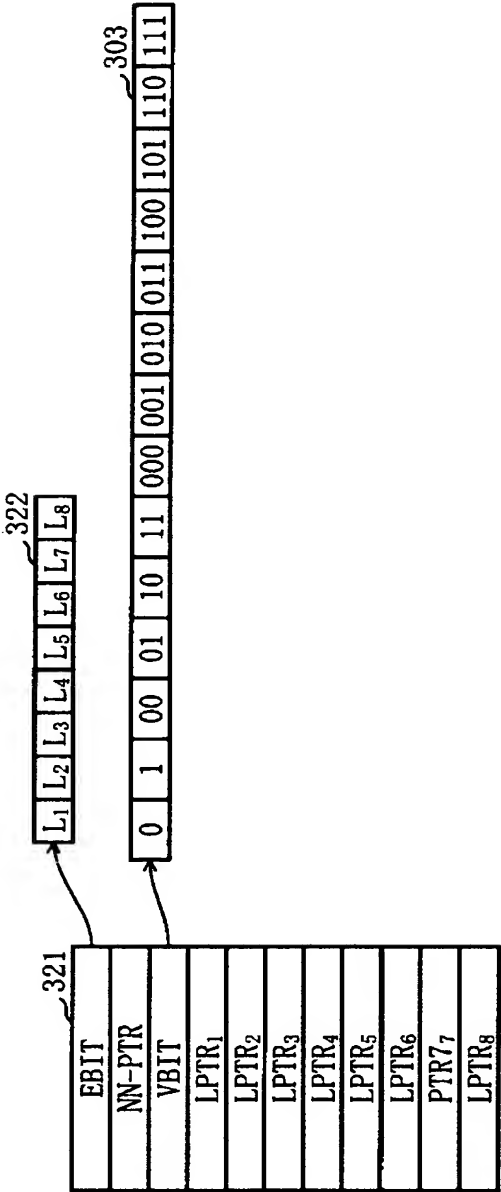


【図 5】

(検索テーブルの内容)

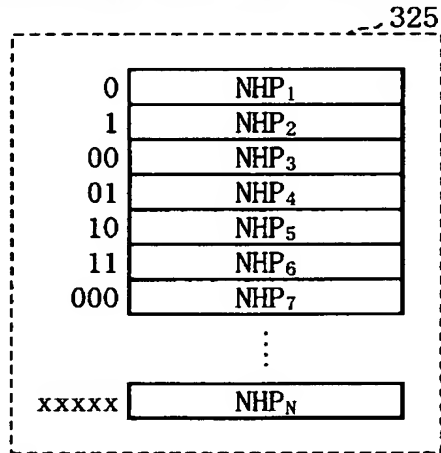


【図 6】



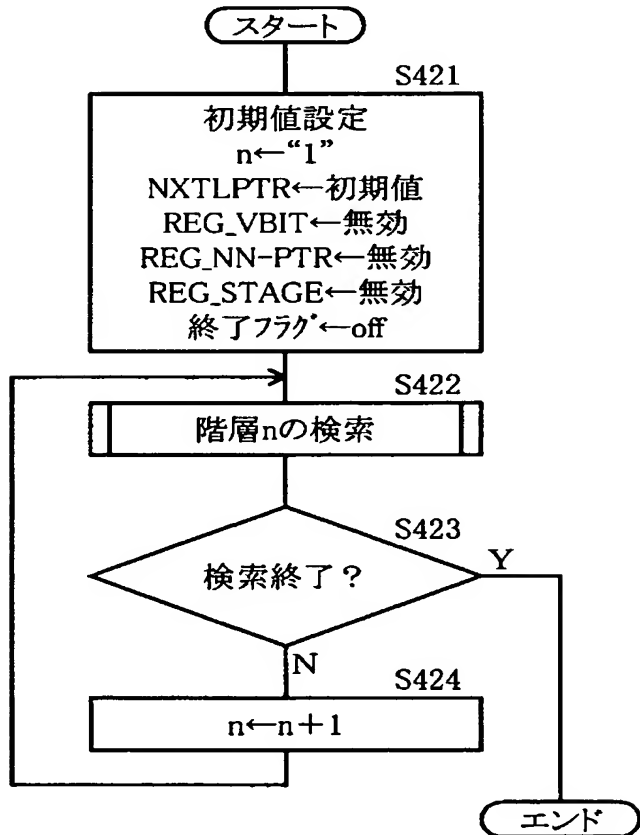
【図 7】

(出力先振り分け情報テーブルの内容)

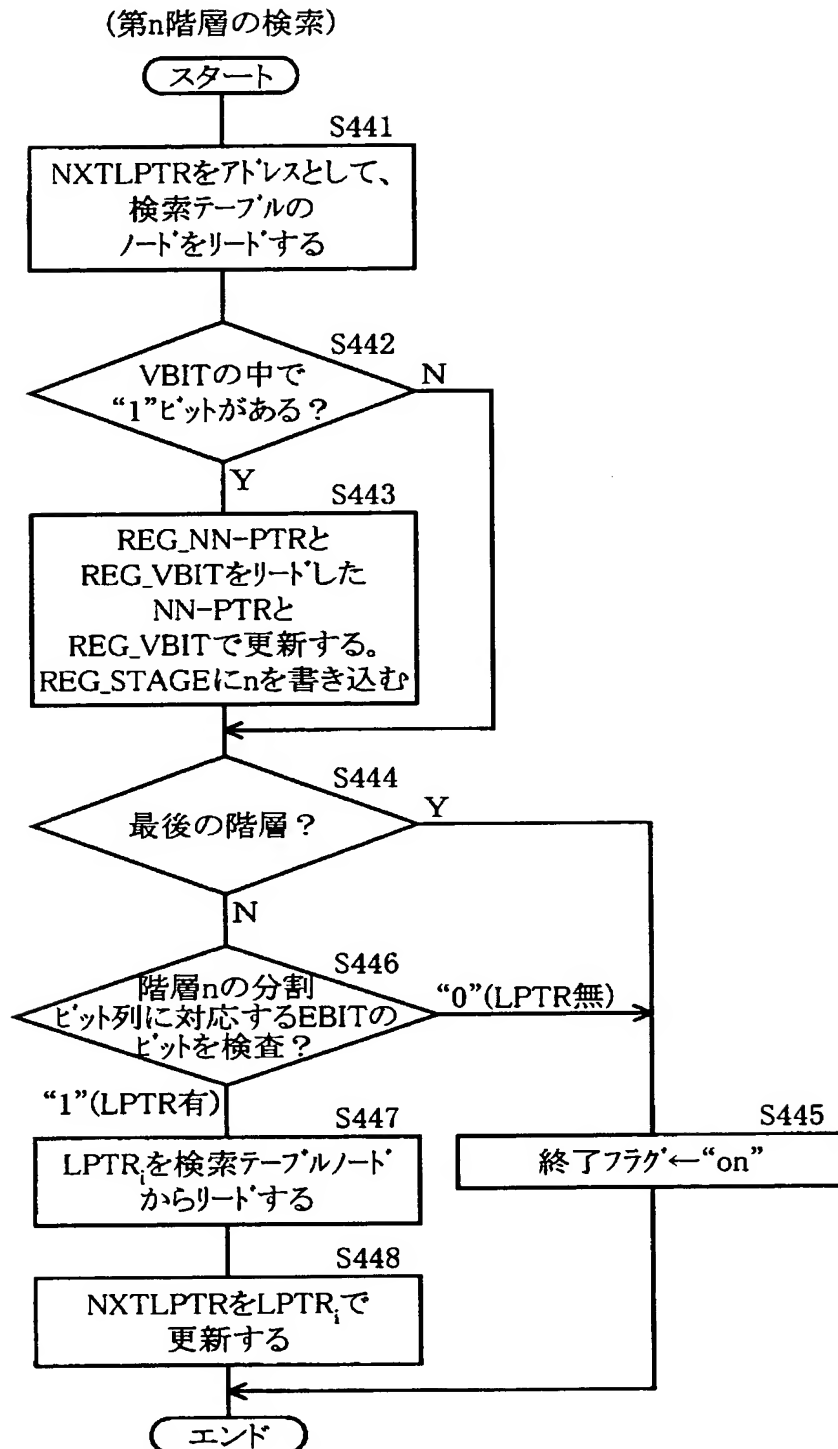


【図 8】

(分割ビット列検索処理)

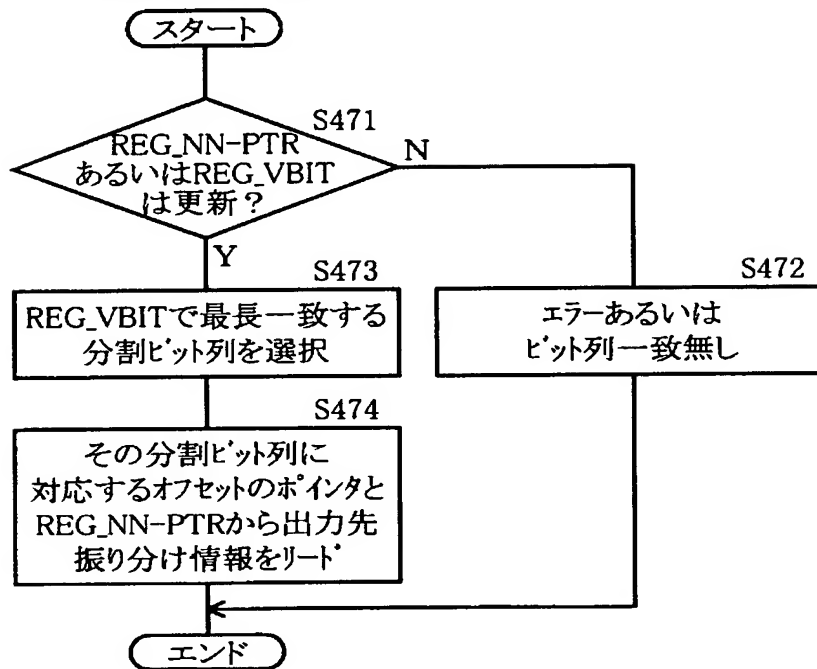


【図 9】

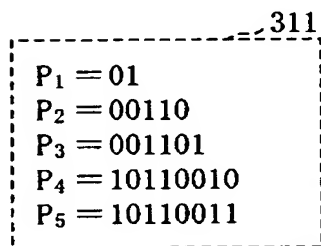


【図 1 0】

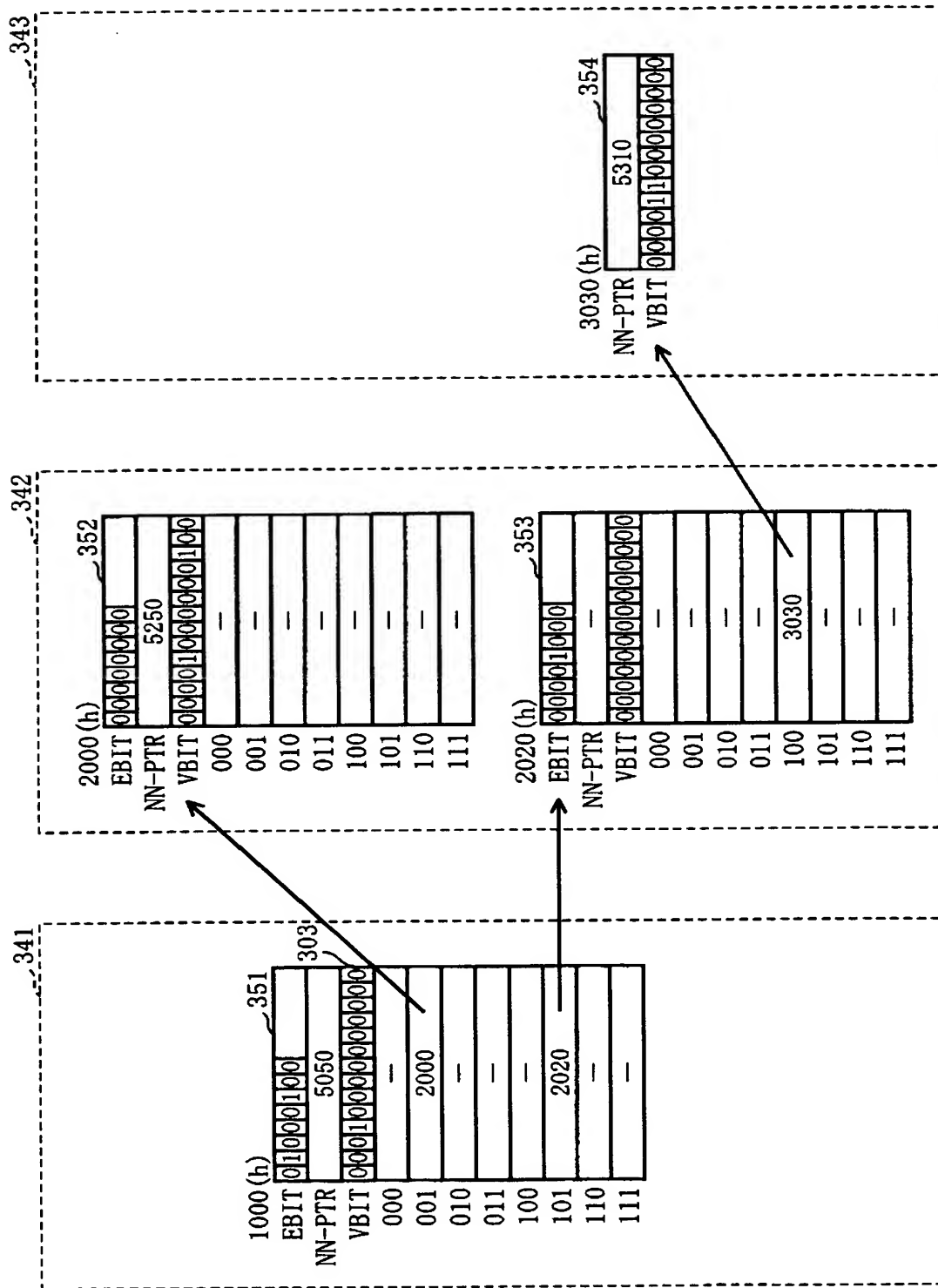
(出力先振り分け情報取得処理)



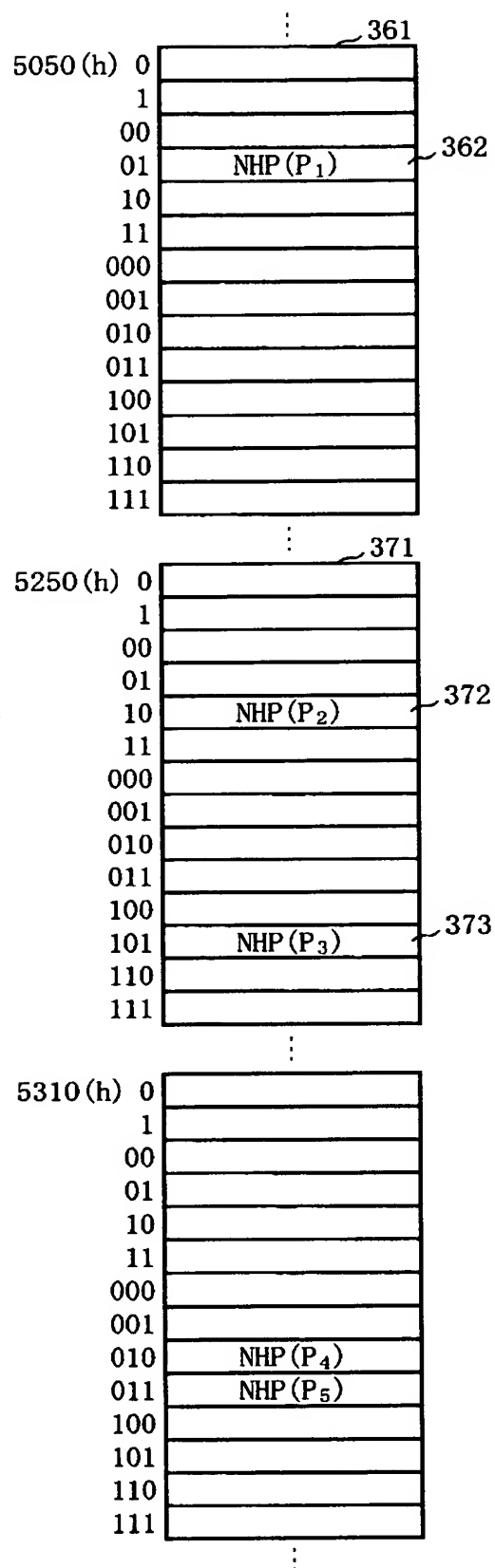
【図 1 1】



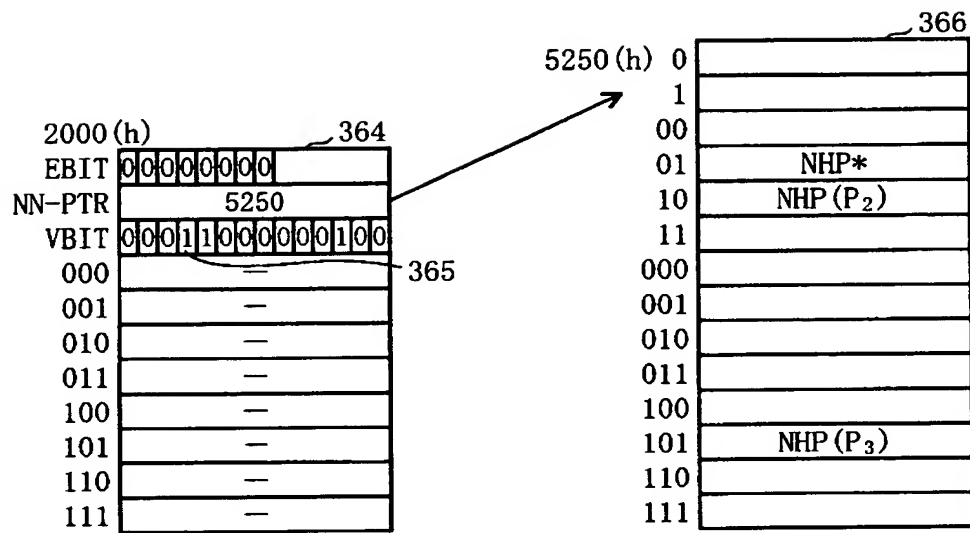
【図 1 2】



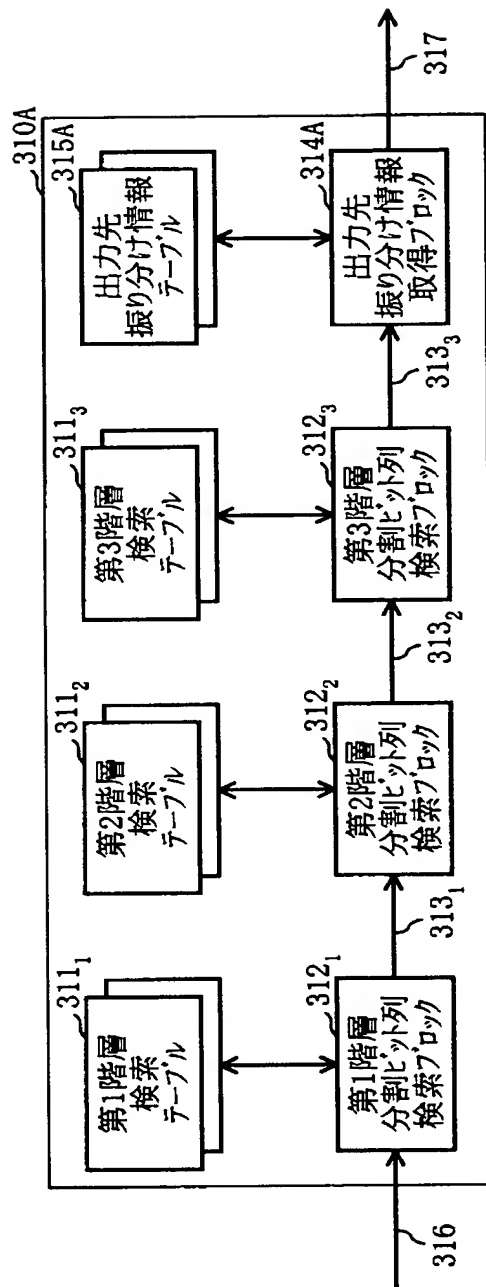
【図 1 3】



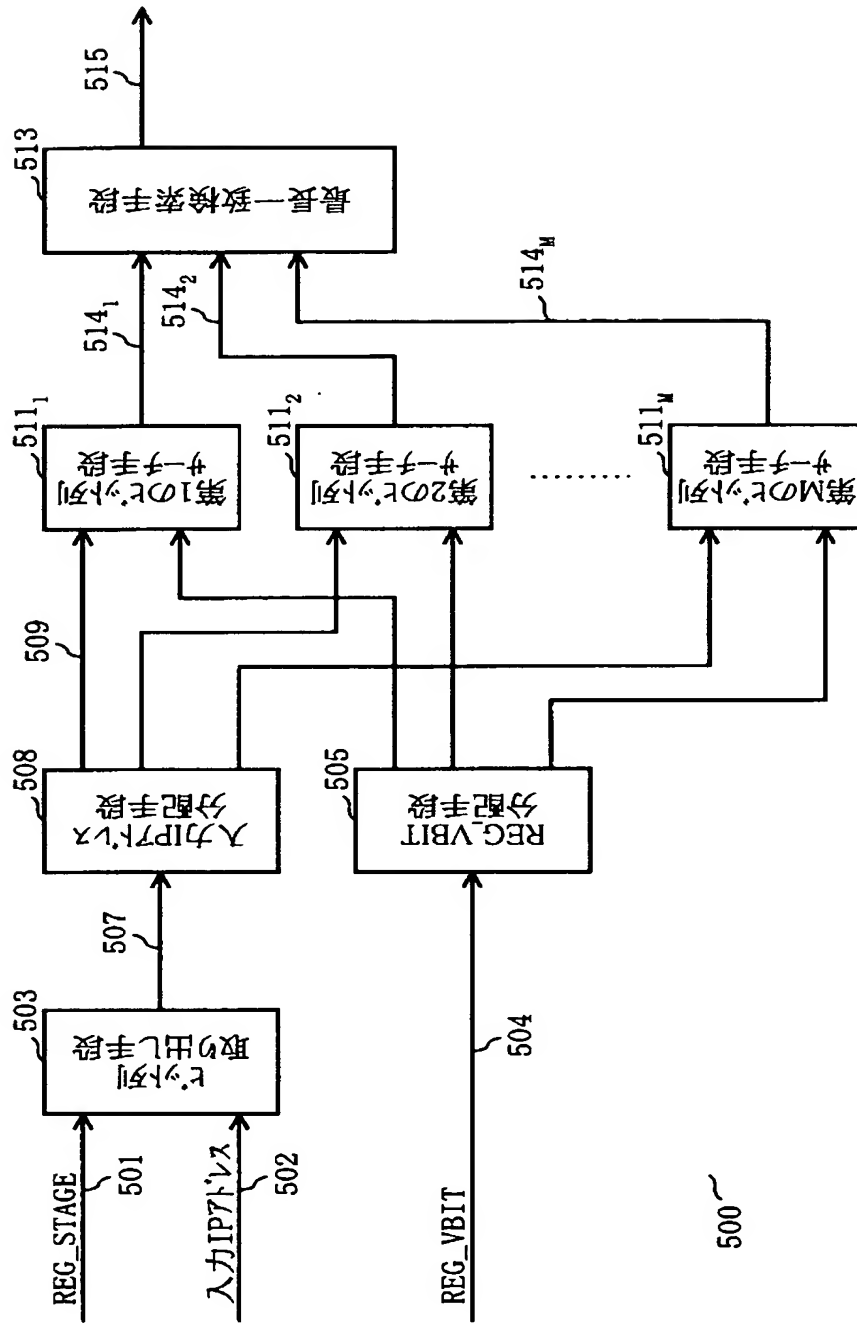
【図 1 4】



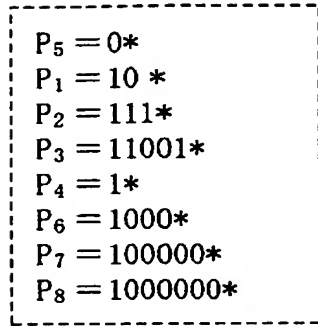
【図 1 5】



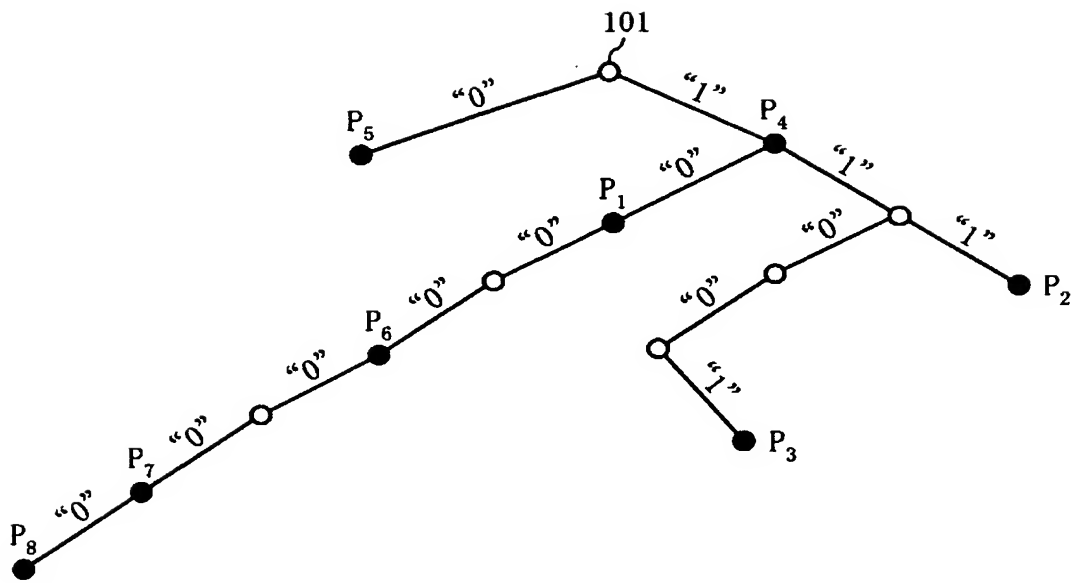
【図 1 6】



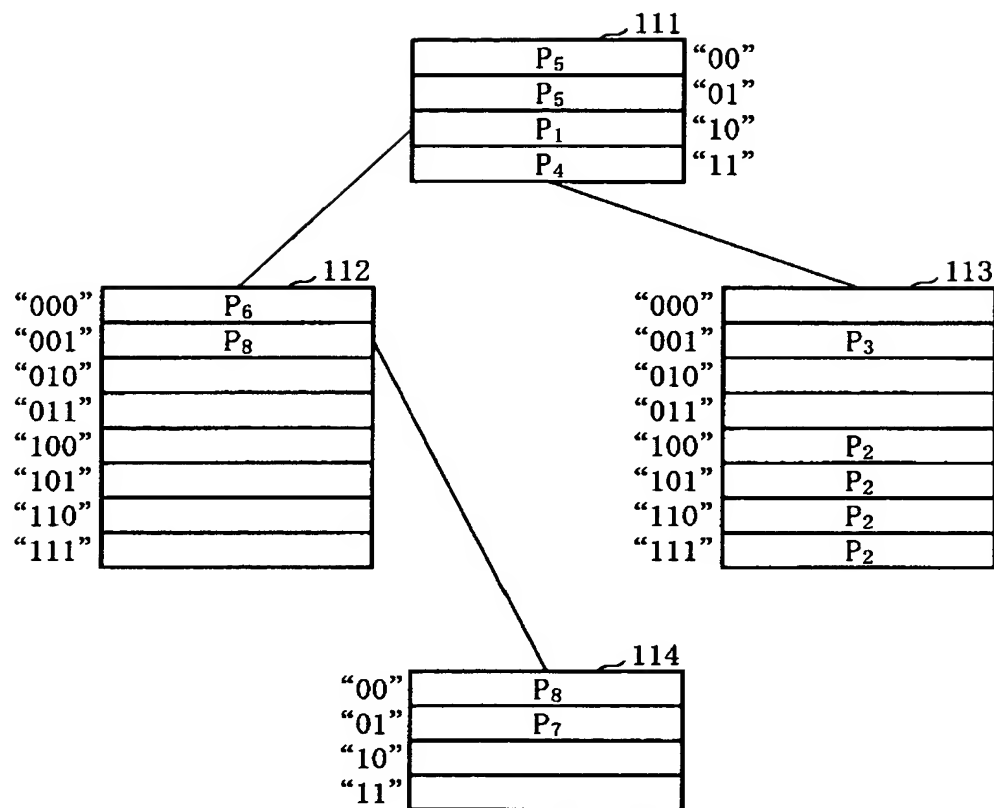
【図 1 7】



【図 1 8】



【図 1 9】



【図 2 0】

Original	Expanded (3Level ₅)	
P ₅ = 0*	00* (P ₅)	Length2
P ₁ = 10 *	01* (P ₅)	
P ₂ = 111*	10* (P ₁)	Length5
P ₃ = 11001*	11* (P ₄)	
P ₄ = 1*	11100* (P ₂)	
P ₆ = 1000*	11101* (P ₂)	
P ₇ = 100000*	11110* (P ₂)	Length7
P ₈ = 1000000*	11111* (P ₂)	
	11001* (P ₃)	
	10000* (P ₆)	
	10001* (P ₆)	
	1000001* (P ₇)	
	1000000* (P ₈)	

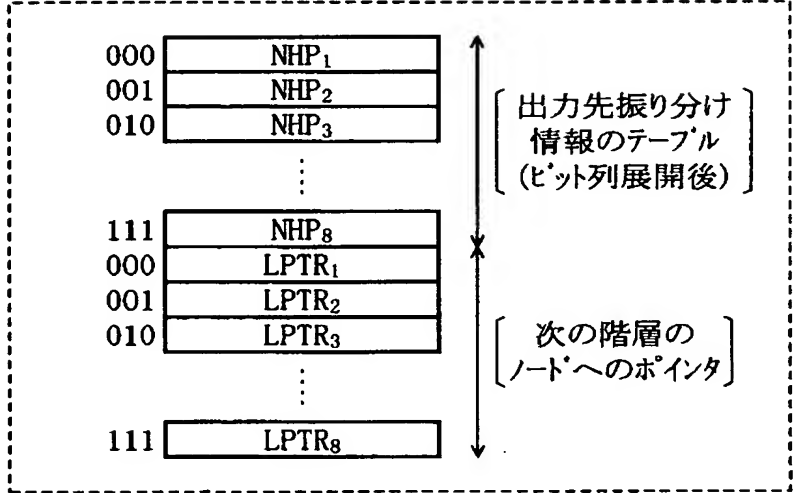
【図 2 1】

$P_1 = 0$
$P_2 = 010$
$P_3 = 1010$
$P_4 = 10$

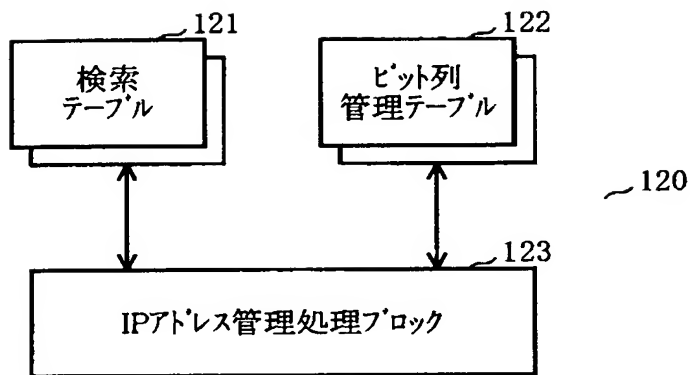
【図 2 2】

"0000"	P_1
"0001"	P_1
"0010"	P_1
"0011"	P_1
"0100"	P_2
"0101"	P_2
"0110"	P_1
"0111"	P_1
"1000"	P_4
"1001"	P_4
"1010"	P_3
"1011"	P_4
"1100"	NULL
"1101"	NULL
"1110"	NULL
"1111"	NULL

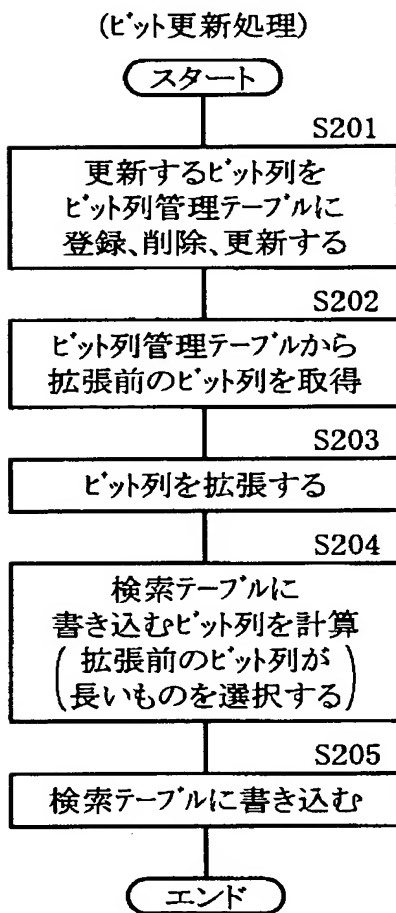
【図 2 3】



【図 2 4】



【図 2 5】



【書類名】 要約書

【要約】

【課題】 通信ネットワークのルーティングを行うための検索テーブルへのアドレスの追加や削除を容易にすること。

【解決手段】 アドレス検索装置 3 1 0 は、分割ビット列の各階層のテーブルを備えた検索テーブル 3 1 1 と、分割ビット列検索ブロック 3 1 2 と、これから得られた検索結果情報 3 1 3 を基にしてデータ信号の振り分けのための情報を取得する出力先振り分け情報取得ブロック 3 1 4 と、出力先振り分け情報テーブル 3 1 5 とで構成されている。IP アドレス 3 1 6 は複数の階層の分割ビット列に順次分解され、先頭から一意に定まるビット列となる階層の検索テーブル 3 1 1 とそのときの分割ビット列との組み合わせによって出力先振り分け情報が読み出される。

【選択図】 図 4

認定・付加情報

特許出願の番号	特願 2 0 0 2 - 2 1 4 0 2 0
受付番号	5 0 2 0 1 0 8 1 7 2 3
書類名	特許願
担当官	第八担当上席 0 0 9 7
作成日	平成 1 4 年 7 月 2 4 日

< 認定情報・付加情報 >

【提出日】 平成14年 7月23日

出 願 人 履 歴 情 報

識別番号 [0 0 0 0 0 4 2 3 7]

1. 変更年月日 1 9 9 0 年 8 月 2 9 日
[変更理由] 新規登録
住 所 東京都港区芝五丁目 7 番 1 号
氏 名 日本電気株式会社